The Thesis Committee for Obehi Georgina Dibua
Certifies that this is the approved version of the following Thesis:

Simulation, Experimentation and Calibration of Nanoparticle Sintering for a Microscale Selective Laser Sintering Process

APPROVED BY
SUPERVISING COMMITTEE:

Michael A. Cullinan, Supervisor

Carolyn C. Seepersad

Simulation, Experimentation and Calibration of Nanoparticle Sintering for a Microscale Selective Laser Sintering Process

by

Obehi Georgina Dibua

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2018

Dedication

To my family.

Acknowledgements

I would like to thank my advisor Dr. Michael Cullinan for introducing me to this project. His advice and support throughout this process has been invaluable. I would like to thank C. S. Foong, Dr. Nilabh Roy, and Dipankar Behera for their advice and intellectual support during the research meetings. I would like to thank Dr. Anil Yuksel for his help with the bed generation simulations. I would also like to thank Joon Cho for his crucial help with the furnace sintering experiments. I would like to thank Raluca Gearba and Hugo Celio over at TMI for their help with the SEM imaging and TGA sintering experiments. This study was made possible through the Supercomputing resources of the Texas Advanced Computing Center. Finally, I would like to thank NSF and NXP Semiconductors for the financial support provided.

Abstract

Simulation, Experimentation and Calibration of Nanoparticle Sintering for a Microscale Selective Laser Sintering Process

Obehi Georgina Dibua, M.S.E.

The University of Texas at Austin, 2018

Supervisor: Michael A. Cullinan

This thesis presents an approach for modeling the sintering of hundreds of 3D nanoparticles in a microscale selective laser sintering process. A phase field modeling approach is utilized in this thesis for studying the densification and microstructural evolution taking place within the nanoparticle bed during sintering. This model is important because there are currently no known simulations that are capable of modeling 3D nanoparticle sintering for over a hundred particles. These simulations are necessary for understanding the effects of sintering parameters on the final properties of sintered parts. From this understanding, input sintering parameters can be varied to create optimum parts. Additionally, post processing algorithms are developed in this thesis for visualizing the evolution of the sintered particles with time and for deriving the densification and shrinkage curves from the data obtained from the sintering simulation. The thesis also discusses in detail the experiments done to provide actual densification curves of copper nanoparticles undergoing sintering. These densification curves are used to validate the simulations and calibrate the simulations to extract sintering constants which cannot be

determined experimentally. Examples of sintering constants are presented and mapped from simulation units to physical units and checked against bulk copper constants. Comparing the results of the simulations to the experiments shows that the trend derived from the densification of the simulation is the same as that derived from the experiments. Also, there is found to be good agreement between the sintering constant derived from the simulations in this study and those found in literature from the sintering of bulk copper. These results validate the phase field modeling approach for evaluating the microstructural evolution of nanoparticle beds during sintering operations and provides us with a valuable tool for modeling and understanding the microscale selective laser sintering process.

Table of Contents

List of Tables

List of Figures

CHAPTER 1: INTRODUCTION

The goal of this project is to model the sintering of nanoparticles for microscale selective laser sintering. This chapter is broken up into three parts, the overview, background and scope.

1.1 Overview

This section presents a broad overview of additive manufacturing, with a focus on microscale selective laser sintering and why this new process is needed. Then modeling this process is introduced and the goals of the project are outlined.

1.1.1 ADDITIVE MANUFACTURING

Additive manufacturing is a process by which parts are created from the ground up. This allows for the creation of parts without the material wastage which occurs in subtractive manufacturing. Additionally, the flexibility of the process allows for the creation of unique parts which are easy to design and rapidly produced. This has made additive manufacturing desirable for industries which rely on the production of customizable parts or rapid prototyping. Some examples of industries which have taken advantage of the benefits of additive manufacturing are the aerospace and medical industries. However, additive manufacturing has minimum feature sizes on the order of hundreds of microns in most commercially available metal additive manufacturing processes [1,2]. This is a shortcoming which has prevented additive manufacturing from expanding into other industries where microscale parts are required, such as the micro-electronics industry. A possible solution to this challenge is through the creation of a microscale selective laser sintering process.

1.1.2 SELECTIVE LASER SINTERING

Selective laser sintering is a powder based additive manufacturing process which uses laser light to heat up and fuse together particles in a powder bed. In the traditional SLS process, powder is spread onto a bed and a laser beam is rastered over the powder bed, providing the particles heated by the laser with enough heat energy to melt and fuse together to form a solid part [3]. This process is different from other AM processes, in that it is powder based and uses a laser as the heat source. The powder base allows for the creation of features like overhangs as sintered parts are supported by unsintered powder underneath it. This process is repeated layer by layer until a full 3D part is made. Though this process gives the desired true 3D structure it has the same challenges in the minimum feature sizes available.

1.1.3 MICROSCALE SELECTIVE LASER SINTERING

Unlike the typical SLS process, microscale selective laser sintering uses the sintering of nanoparticles to give a better control on feature sizes [4-6]. Similar to the SLS system, the microscale selective laser sintering process is a layer by layer process which uses the selective sintering of each layer to create a true 3D structure. The sintering base in microscale selective laser sintering is called a substrate. This substrate is placed on top of a heated chuck on a nanopositioning stage. Each layer of nanoparticles is deposited onto the substrate using a slot die coater. The slot die coater deposits layers of nanoparticle inks instead of nanoparticle powders because the powders agglomerate without the solvent in the nanoparticle ink. The substrate is shuttled between the slot die coating system and the optical system using a linear servo motor. The nanopositioning stage allows for a finer

control on the positioning of the substrate than is allowed with the linear servo motor. The optical system is where the laser beam is introduced into the substrate and the parts are sintered. This system is currently being built and studied at the University of Texas at Austin and a design is shown in Figure 1.1. With the creation of this process, there is a concurrent need for the creation of models to help predict the final properties of sintered parts.

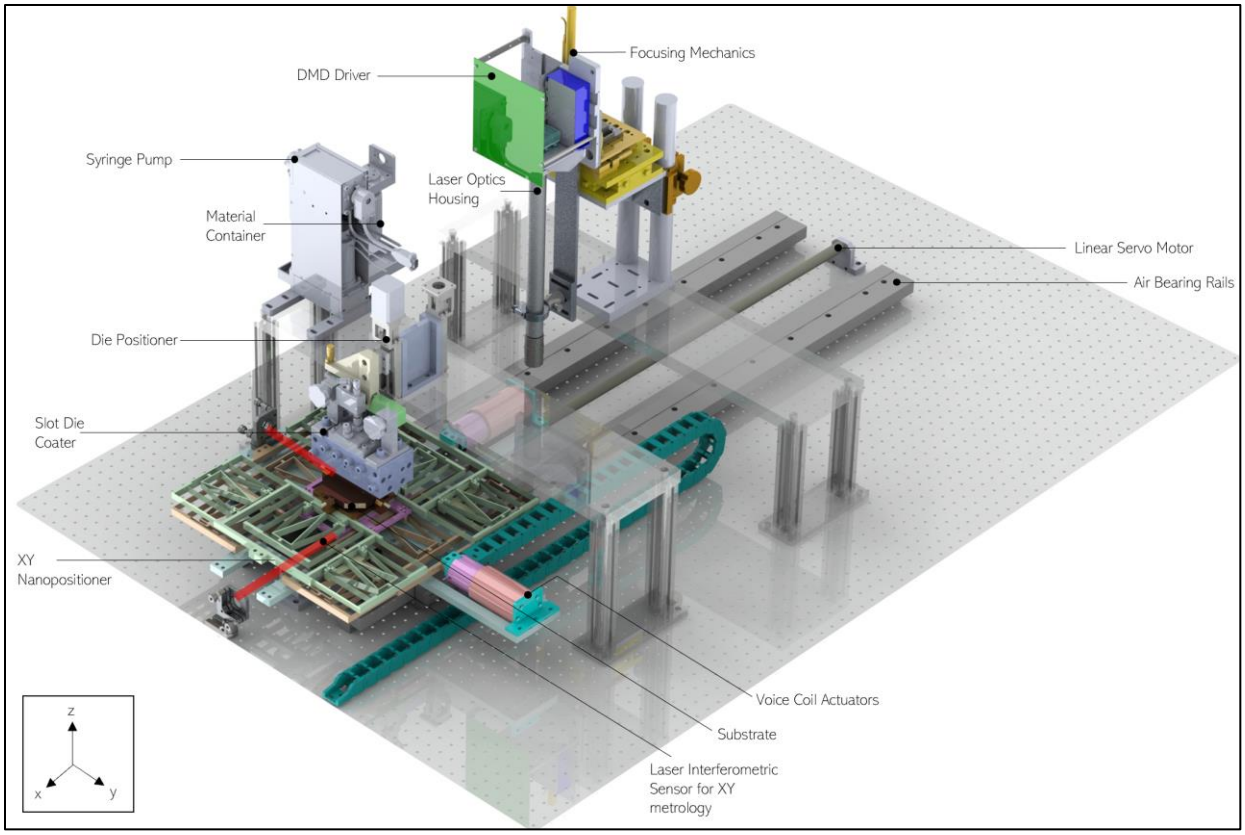


Figure 1.1: Design of Microscale Selective Laser Sintering System

1.1.4 PROJECT GOAL

There are currently a number of models which study the sintering of particles in an SLS process [7-9]. These simulations model sintering as a melting process. While this is accurate for microparticles, when particles tend to the nanoscale the forces driving sintering changes. As such there is a need for models which are able to account for the nanoscale effects taking place during sintering.

There are a number of processes which take place during the sintering of nanoparticles, some of which include vapor deposition, twinning, grain boundary dislocation, viscous flow and diffusion [10]. The particles used for this study are on the order of ten to a hundred nanometers. Researchers have found that for this size of particles, the sintering process is dominated by surface and then grain boundary diffusion [10]. As such, the most efficient model used for this project would be a diffusion model. The goal of this project is to model the diffusion driven sintering between a large number of nanoparticles. A large number of nanoparticles for this study is quantified as being over a hundred particles.

1.2 Background

In this section alternative modelling approaches are discussed. Studying sintering as a diffusion process is not a novel idea. A number of simulations study sintering as a diffusion driven process. Four approaches used will be discussed here. These approaches are the Discrete Element Method (DEM), Molecular Dynamics (MD), Monte Carlo Method and Phase Field Modelling (PFM). A comparison is done between these modeling approaches and PFM is chosen. The differences between the PFM simulation discussed in this thesis and the PFM simulations which already exist are highlighted.

1.2.1 DISCRETE ELEMENT METHOD (DEM)

In the discrete element method, particles in a system are treated as distinct spheres interacting with each other. Prior to the onset of sintering this method is able to fully capture the particle contact interactions using models like the Hertzian or linear spring damping collisions. However, once sintering starts the assumption of distinctness between particles no longer holds true as the particle diffuse into each other. Rojek et al modeled sintering as a diffusion process using a Discrete Element Method (DEM) [11]. In this study an explicit force function describing the sintering behavior between particles is required. The final bed configuration (after sintering) presented by Rojek et al has particles as discrete spheres, highlighting the distinctness problem previously mentioned. The upside to using DEM is that it is less computationally expensive than the other approaches discussed and as such can model particles on a bed scale. However, DEM makes the assumption that particles remain spherical throughout sintering which gives the disadvantage of inaccurate data and invalid sintered particle morphologies.

1.2.2 MOLECULAR DYNAMICS (MD)

Molecular Dynamics simulations track the evolution of a particle system by monitoring the interactions between atoms. In these models, particles are made up of a number of atoms and the migration of these atoms determine the sintering kinetics moving from one time point to another. These models are very detailed, as they are able to track all the nanoscale effects taking place during sintering and have been used to determine material properties which are difficult to get from experiments [12]. However, the downside to using MD simulations is the computational expense associated with running them. Because MD simulations track the interactions between atoms, these simulations are only able to model a few particles at a time.

Ding and Pan studied the sintering between nanoparticles using molecular dynamics (MD) simulations [13]. These MD simulations made use of the Lennard-Jones model to track the potential between particles. Ding and Pan also compared continuum and MD models for the sintering of nanoparticles. This comparison led to the realization that the preliminary assumptions required for continuum models made them inadequate for tracking the sintering kinetics which change during the MD simulations. Li and Pan studied the differences between the sintering of micro- and nano-particles using MD simulations [14]. They found that the main difference was in the coarsening process. An example of this is highlighted in a 3-particle system where two larger particles surround a small particle. In the micro particles, because sintering is dominated by grain boundary diffusion the grain boundaries from the larger particles move rapidly towards the smaller particle. In nanoparticles, since sintering is dominated by surface diffusion, the atoms on the surface of the larger particles move towards the smaller particle faster than the grain boundary moves. Cheng and Ngan were able to expand from the few particle MD simulations to tracking four particles, broken into half, quarter and octet portions and arranged in an FCC crystalline state [15]. Cheng and Ngan improved on the number of particles modeled by placing a single particle in the center of a simulation box and having smaller portions of other particles surrounding it. But even in this case, the higher number of particles modeled does not come close to the bed scale requirement for this study.

1.2.3 MONTE CARLO

Monte Carlo method is a stochastic approach. In this method, a simulation space is broken up into different lattice sites and these sites are tagged by degenerate states. Depending on the value of the degenerate state a lattice has, the lattice is considered to

either be in one of the particles (solid phase) or in a pore (vapor phase). The evolution of sintering in this model is based off a random selection and comparison of lattice sites. First a lattice site is chosen at random and then a neighboring site is chosen at random. New degenerate states are assigned to these lattice sites randomly (but selection rules apply based off the current degenerate states). The goal is to consider if moving from the old to the new cite minimizes the total energy. If energy is minimized, the sites move, if not a standard Metropolis algorithm is used. In this algorithm, a random number R between 0 and 1 is generated and compared to a value P defined by the Boltzmann statistics. If R is less than or equal to P the change is accepted, if not the previous state is restored. The upside to using this method is that it is less computationally expensive than molecular dynamics and that it does not require the knowledge of any material properties. The lack of material properties in running this simulation is also a downside because the simulation cannot then be used to predict properties which cannot easily be found using experiments.

Braginsky et al [16] developed a Kinetic Monte Carlo approach to simulate solid-state sintering, and then used the microstructural evolution obtained to determine the sintering stress. In this work, a vacancy annihilation scheme was utilized. In this scheme, vacancies at the grain boundaries are removed by exchanging those lattice sites with solid sites at the surface. The results are compared against other models and the importance of undefined geometries are shown. The authors set forth three requirements for the realistic simulation of microstructural evolution. The first is that model should not depend on the particles being a specific geometry, that is, the model should still be valid if during the course of sintering the geometry of particles change. Second is that the model should not rely on the division of the sintering process to specific stages with a pre-defined start time, as it is impossible to state generically when one stage would start and another ends. Finally,

the model should be scalable. Chen et al [17] used a similar approach to study densification during solid-state sintering. In this study densification is achieved using a penalty energy which enhances the minimization of the void-particle surface energy. This approach is shown to yield much better results, in agreement with expected experimental trends, than the standard Monte Carlo simulation. Bjork et al [18] studied the effect of particle size distribution on microstructural evolution during sintering, using the monte carlo method. They found that in broader distributions there was a higher initial grain growth rate which they attributed to the prevalence of smaller particles, once these particles diffuse into the larger grains, the sintering stalls leading to lower final densities.

1.2.4 PHASE FIELD MODELING (PFM)

Phase Field Modeling is a deterministic approach. In this modeling approach a simulation box is discretized into pixel grids and each pixel is assigned two variables. These variables differentiate between the solid and vapor phase, as well as different particles in the system. The evolution of sintering using this method is governed by the numerical solution of partial differential equations which relate the rate of change in the phase field variables. The explicit form of these equations is discussed in more detail in the modeling section. Similar to the Monte Carlo method, the diffusion in the PFM model is driven by the minimization of the total energy in the system. Unlike the Monte Carlo method, PFM is not stochastic. Also, the PFM model makes use of constants related to material properties. While this requires for the constants to be set in order to run the simulation, adding an additional level of complexity that does not exist in the Monte Carlo method, it also gives the possibility of determining material properties from the simulation.

A number of Phase Field models have been created to track the sintering behavior between particles. Wang used PFM to track the sintering between 26 particles [19] and Shinagawa used a combination of PFM and DEM to track the sintering between a cluster made up of 10 particles [20]. With both of these models, the sintering behavior is successfully characterized for the particles in 2D. While 2D simulations provide useful insight into the process of diffusion between particles, they cannot accurately be used to predict properties of real particle beds such as volumetric shrinkage. Kumar used PFM to track the sintering of 28 3D particles randomly arranged in a cubic lattice [21]. Chickalingam et al [22] used Phase Field Modeling to simulate the sintering of silver nanoparticles. This work showed the viability of using PFM for simulating the microstructural evolution of nanoparticles during sintering and validated the results against experimental data for two 30 nm particles.

Tikare et al [23] compared the phase field model and Monte Carlo model during grain growth and Ostwald ripening for many particles in 2D. They found that during both of these processes the quantitative results from PFM and MC were very similar and within statistical errors. The main differences where qualitative, where the PFM grains were found to be regular and smooth while the MC grains were irregular and equiaxed. This difference is due to the modeling approaches as PFM utilizes a diffuse interface and MC is a sharp interface model.

1.2.5 Comparison between Molecular Dynamics, Monte Carlo and Phase Field Modeling approaches

The MD, Monte Carlo and PFM simulations were compared based on three criteria:

i. Scalability to nanoscale sintering

ii. Computational expense

iii. Ability to derive physical constants from simulation (Physical constants)

These criteria were chosen as the most valid in relation to the use of the modeling approaches for this project. Scalability to nanoscale sintering is mandatory as the goal of the project is to account for the sintering of nanoparticles. Computational expense is important in order to have the ability to model hundreds of particles in 3D. Deriving physical constants, such as diffusion coefficients and energy variables, from the simulation is important because there are no known ways of deriving the constants used in these simulations experimentally or analytically. The comparison is summarized in Table 1.1. The scaling for this comparison goes between 0 and 2. A value of 0 is given if a criterion is not satisfied, a value of 1 is given if extensive modification is required to make a criterion satisfied, and a value of 2 is giving if very little modification is needed to make a criterion satisfied.

Approach	Scalability to Nanoscale	Computational expense	Physical Constants		Total
Molecular Dynamics (MD)	2	0	2		4
Monte Carlo (MC)	1	2	0		3
Phase Field Modeling (PFM)	2	2	1		5

Table 1.1: Comparison of Modeling approaches

In terms of computational expense, the MD simulation performs a lot worse than MC and PFM. As discussed earlier, MD simulations cannot be used to model more than a few particles at a time. However, MD approach soars ahead in scalability to sintering nanoparticles and deriving physical constants. However, the inability to perform large scale

sintering with MD, removes it from consideration. The Monte Carlo simulations perform a lot better in terms of computational expense. However, tailoring the Monte Carlo simulation to nanoscale sintering would require adopting a scheme in exchanging lattice sites, which favors the migration of surface lattice sites at a faster rate than diffusion along the grain boundary. While this is not impossible it is not as easy to set as the PFM or MD simulations which is why MC receives a point of 1 here. As discussed earlier, MC simulations do not make use of physical constants related to sintering and as such cannot be used for deriving these constants from simulations. On the other hand, PFM simulations make use of sintering constants and as such these constants can be mapped from simulation constants to physical constants. The process of deriving this is not as straight-forward as MD simulations as such PFM receives a score of 1. In terms of scalability, the use of sintering constants, specifically the surface and grain boundary energy, allow for the amount of surface diffusion to be increased easily, simply by changing these simulation constants. PFM scores the highest total points and as such, PFM was chosen as the modeling approach used for this project.

1.2.6 DIFFERENCE IN PFM SIMULATION

As mentioned in the Phase Field Modeling section, PFM has been successfully used to simulate sintering by a number of authors. In contrast to the previously mentioned models, the simulation used in this project is able to track the sintering between hundreds of particles. The simulation of so many particles is made possible through parallel computing. This project simulates sintering in a number of 2D systems and for 2 particles in 3D, a 43 particle one-by-one micrometer bed, and a 134 particle two-by-two micrometer bed. The next chapter discusses the details of the PFM model used in this project,

introducing the governing equations and discretizing these equations to the forms used in the simulations.

1.3 Scope

This thesis discusses the creation of a model which simulates the sintering of nanoparticles for a microscale selective laser sintering process. The model uses Phase Field Modeling to track the diffusion between the particles in this system. A post processing script is built in python to extract data from the results of the phase field model. This gives the densification and shrinkage curve from the simulation results. The thesis also presents experiments done to validate and calibrate the simulation. These experiments were done using two techniques, Thermogravimetric Analysis and furnace heating. The experiments were done at different temperatures and for different time durations. Density curves were derived from the results of the experiments and used to calibrate the simulation constants. Additionally, simulation constants are mapped from simulation units to physical units to further verify the results of the simulation. The process followed for the creation of these models is discussed in more detail in the following sections and the outline is discussed next.

In the methodology section, the simulation set up is discussed. All the equations used for the simulation are presented here. The experimental procedure is discussed in the following chapter. The set up for the experiments as well as the materials and devices used are described. Finally, the calibration process between the simulation and experiments is described. In the results chapter, the results for the simulation and experiments, described in the methodology chapters, are presented and discussed. The mismatch between the sizes of the simulation bed and experiment samples is also discussed. Then the calibration results

are presented. In the conclusions chapter, the work of the thesis is summarized, and the future work is enumerated and discussed.

CHAPTER 2: MODEL METHODOLOGY[1]

This chapter presents the methodology for deriving the equations used for the sintering simulation as well as the steps for running the simulation. The chapter is broken up into five parts, PFM model setup, PFM simulations, bed generation, post processing with python and a summary of the model methodology.

2.1 PFM Model Setup

In this section the PFM modeling approach is discussed in more detail. The partial differential equations used in this model are presented and the process of discretizing these equations is shown.

2.1.1 PFM MODEL

2.1.1.1 Introduction

Phase Field Modeling (PFM) is a diffuse-interface approach which tracks the evolution of particles using phase field variables which are related to microscopic parameters [24]. In this study, the PFM variables used are the conserved mass density (ρ) variable and the non-conserved order parameter (η_i). These variables take on values from 0 to 1 tracking the phases of, and interphases between, particles. The density variable differentiates between solid phase, where it takes the value of 1, and the vapor phase, where it takes the value of 0. It has values between 0 and 1 at the surface boundaries between the solid and vapor phase. On the other hand, the order parameter takes on the value of 1 for the ith particle and is 0 for every other particle. The order parameter takes on values between 0 and 1 at the grain boundaries between different particles.

[1] Dibua OG, Yuksel A, Roy NK, Foong CS, Cullinan M. Nanoparticle Sintering Model: Simulation and Calibration Against Experimental Data. ASME. J. Micro Nano-Manuf. 2018;6(4):041004-041004-9. doi:10.1115/1.4041668.

2.1.1.2 PFM equations

The evolution of particles in this system is driven by the minimization of the total free energy, which is a function of the phase field variables and is given in Eq. 1.

$$F = \int_V \left[f(\rho, \eta_i) + \frac{1}{2}\beta_\rho |\nabla \rho|^2 + \frac{1}{2}\sum_{i=1}^{N} \beta_\eta |\nabla \eta_i|^2 \right] dV \quad (1)$$

$f(\rho,\eta_i)$ is the bulk free energy, N is the total number of particles in the system, β_ρ is the gradient energy term for the density variable, and β_η is the gradient energy term for the order parameter. The bulk free energy is a Landau type potential shown in Eq. 2. [21].

$$f(\rho, \eta_i) = K_1 \left(\rho^4 + \frac{-4\rho_{vap} - 4\rho_{sol} - 2}{3}\rho^3 + \frac{4\rho_{vap}\rho_{sol} + 2\rho_{vap} + 2\rho_{sol}}{2}\rho^2 - 2\rho_{vap}\rho_{sol}\rho \right) + K_2 \left(\rho^2 + 6(1-\rho)\sum_{i=1}^{N} {\eta_i}^2 - 4(2-\rho)\sum_{i=1}^{N} {\eta_i}^3 + 3\sum_{i=1}^{N} {\eta_i}^4 + w\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} {\eta_i}^2 {\eta_j}^2 \right) \quad (2)$$

where ρ_{vap} and ρ_{sol} are the vapor and solid density parameters respectively, w is related to the grain boundary energy, K_1 and K_2 are constants related to grain boundary and surface energy as shown in Eqs 3 and 4. In addition to K_1 and K_2, the gradient energy terms for the density variable and the order parameter, from Eq. 1, are related to the surface energy and grain boundary energy in the system. Chockalingam et al [22], showed that the surface and grain boundary energy are functions of the constants in Eq. 1 and Eq. 2. The relationship between these constants and the energy terms are shown in Eqs. 3 and 4.

$$\gamma_{gb} = \frac{2\sqrt{K_2 \beta_\eta}}{\sqrt{3}} \quad (3)$$

$$\gamma_s = \frac{\sqrt{2\left(\beta_\rho + \beta_\eta\right)\left(K_1 + 7K_2\right)}}{6} \quad (4)$$

γ_{gb} in Eq. 3 is the grain boundary energy and γ_s in Eq. 4 is the surface energy. As stated earlier, the sintering kinetics are driven by the progression of the phase field variables. The temporal evolution of the density variable is governed by the Cahn-Hilliard equation [25,26] which is shown in Eq. 5.

$$\frac{\partial \rho}{\partial t} = \nabla.\left(D\nabla\frac{\delta F}{\delta \rho}\right) = \nabla.\left(D\nabla\left[\frac{\partial f(\rho,\eta_i)}{\partial \rho} - \beta_\rho \nabla^2 \rho\right]\right) \quad (5)$$

D is the equivalent diffusion coefficient, made up of fractions of the surface, grain boundary and volume diffusion coefficients, and is given by Eq. 6.

$$D = D_{vol}\emptyset + D_{surf}\rho^2(1-\rho^2) + D_{gb}\rho\left(1 - \sum_{i=1}^{N}{\eta_i}^2\right) \quad (6)$$

D_{vol} is the volume diffusion coefficient, D_{surf} is the surface diffusion coefficient, D_{gb} is the grain boundary diffusion coefficient, and $\emptyset = \rho^4(7\rho^2 - 18\rho + 12)$ [21] having a maximum in the solid phase and a minimum in the vapor phase. The temporal evolution of the order parameter is governed by the time-dependent Ginzburg-Landau structural relaxation equation [27], given in Eq. 7 where L is a constant that characterizes grain boundary mobility.

$$\frac{\partial \eta_i}{\partial t} = -L\frac{\partial F}{\partial \eta_i} = -L\left(\frac{\partial f}{\partial \eta_i} - \beta_\eta \nabla^2 \eta_i\right) \quad (7)$$

2.1.2 DISCRETIZED PFM EQUATIONS

The Taylor series approximation of a function f(x) at some value a, is given in Eq. 8.

$$f(a) = f(x) + f'(x)(a-x) + \frac{f''(x)}{2}(a-x)^2 + \frac{f'''(x)}{6}(a-x)^3 + \cdots + \frac{f^{(n)}(x)}{n!}(a-x)^n \quad (8)$$

Where a = x ± Δx. Neglecting higher order terms (3rd order and higher) for small step sizes (small Δx) and expanding to three dimensions for x,y and z, the equation above becomes Eq. 9.

$$\begin{aligned} f(x+\Delta x, y+\Delta y, z+\Delta z) \\ &= f(x,y,z) + f_x(x,y,z)\Delta x + f_y(x,y,z)\Delta y + f_z(x,y,z)\Delta z + \frac{f_{xx}(x,y,z)}{2}(\Delta x)^2 \\ &+ \frac{f_{yy}(x,y,z)}{2}(\Delta y)^2 + \frac{f_{zz}(x,y,z)}{2}(\Delta z)^2 + f_{xy}(x,y,z)\Delta x\Delta y + f_{xz}(x,y,z)\Delta x\Delta z \\ &+ f_{yz}(x,y,z)\Delta y\Delta z \end{aligned} \quad (9)$$

Looking separately at the gradients in x, (Δy = Δz = 0), the Eq. 9 becomes Eqs. 10 and 11.

$$f(x+\Delta x, y, z) = f(x,y,z) + f_x(x,y,z)\Delta x + \frac{f_{xx}(x,y,z)}{2}(\Delta x)^2 \quad (10)$$

$$f(x-\Delta x, y, z) = f(x,y,z) - f_x(x,y,z)\Delta x + \frac{f_{xx}(x,y,z)}{2}(\Delta x)^2 \quad (11)$$

Adding both equations together and simplifying to solve for the second derivative in terms of x gives Eq. 12

$$f_{xx}(x,y,z) = \frac{f(x+\Delta x, y, z) + f(x-\Delta x, y, z) - 2f(x,y,z)}{(\Delta x)^2} \quad (12)$$

Similarly, the second derivatives in y and z are listed in Eqs. 13 and 14.

$$f_{yy}(x,y,z) = \frac{f(x, y+\Delta y, z) + f(x, y-\Delta y, z) - 2f(x,y,z)}{(\Delta y)^2} \quad (13)$$

$$f_{zz}(x,y,z) = \frac{f(x, y, z+\Delta z) + f(x, y, z-\Delta z) - 2f(x,y,z)}{(\Delta z)^2} \quad (14)$$

Relating this back to equation 5, the Laplace operator acting on ρ is defined in Eq. 15.

$$\nabla^2\rho = \rho_{xx} + \rho_{yy} + \rho_{zz} = \frac{\rho(x+\Delta x,y,z) + \rho(x-\Delta x,y,z) - 2\rho(x,y,z)}{(\Delta x)^2} + \frac{\rho(x,y+\Delta y,z) + \rho(x,y-\Delta y,z) - 2\rho(x,y,z)}{(\Delta y)^2} + \frac{\rho(x,y,z+\Delta z) + \rho(x,y,z-\Delta z) - 2\rho(x,y,z)}{(\Delta z)^2} \quad (15)$$

If a new term h is defined such that

$$h = \frac{\partial f}{\partial \rho} - \beta_\rho \nabla^2 \rho \quad (16)$$

Then from equation 2, the partial derivative of *f* with respect to ρ is given as in Eq. 17.

$$\frac{\partial f}{\partial \rho} = K_1\left(4\rho^3 + \left(-4\rho_{vap} - 4\rho_{sol} - 2\right)\rho^2 + \left(4\rho_{vap}\rho_{sol} + 2\rho_{vap} + 2\rho_{sol}\right)\rho - 2\rho_{vap}\rho_{sol}\right) + K_2\left(2\rho - 6\sum_{i=0}^{N}{\eta_i}^2 + 4\sum_{i=0}^{N}{\eta_i}^3\right) \quad (17)$$

Combining equations 15, 16 and 17, h is defined as in Eq. 18.

$$h = K_1\left(4\rho^3 + \left(-4\rho_{vap} - 4\rho_{sol} - 2\right)\rho^2 + \left(4\rho_{vap}\rho_{sol} + 2\rho_{vap} + 2\rho_{sol}\right)\rho - 2\rho_{vap}\rho_{sol}\right) + K_2\left(2\rho - 6\sum_{i=0}^{1}{\eta_i}^2 + 4\sum_{i=0}^{1}{\eta_i}^3\right) - \beta_\rho\left(\frac{\rho(x+\Delta x,y,z) + \rho(x-\Delta x,y,z) - 2\rho(x,y,z)}{(\Delta x)^2} + \frac{\rho(x,y+\Delta y,z) + \rho(x,y-\Delta y,z) - 2\rho(x,y,z)}{(\Delta y)^2} + \frac{\rho(x,y,z+\Delta z) + \rho(x,y,z-\Delta z) - 2\rho(x,y,z)}{(\Delta z)^2}\right) \quad (18)$$

Using Eq. 16, Eq. 5 can be rewritten as Eq. 19.

$$\frac{\partial \rho}{\partial t} = \nabla.\left(D\nabla\frac{\delta F}{\delta \rho}\right) = \nabla.\left(D\nabla\left[\frac{\partial f(\rho,\eta_i)}{\partial \rho} - \beta_\rho\nabla^2\rho\right]\right) = \nabla(D\nabla h) = \nabla D\nabla h + D\nabla^2 h \quad (19)$$

where the individual terms are given in Eqs. 20 and 21.

$$\nabla D\nabla h = \frac{D(x+\Delta x,y,z)[h(x+\Delta x,y,z)-h(x,y,z)]}{\Delta x^2} - \frac{D(x-\Delta x,y,z)[h(x,y,z)-h(x-\Delta x,y,z)]}{\Delta x^2} + \frac{D(x,y+\Delta y,z)[h(x,y+\Delta y,z)-h(x,y,z)]}{\Delta y^2} - \frac{D(x,y-\Delta y,z)[h(x,y,z)-h(x,y-\Delta y,z)]}{\Delta y^2} + \frac{D(x,y,z+\Delta z)[h(x,y,z+\Delta z)-h(x,y,z)]}{\Delta z^2} - \frac{D(x,y,z-\Delta z)[h(x,y,z)-h(x,y,z-\Delta z)]}{\Delta z^2} \quad (20)$$

$$D\Delta^2 h = D(x,y,z)\left(\frac{h(x+\Delta x,y,z)+h(x-\Delta x,y,z)-2h(x,y,z)}{(\Delta x)^2} + \frac{h(x,y+\Delta y,z)+h(x,y-\Delta y,z)-2h(x,y,z)}{(\Delta y)^2} + \frac{h(x,y,z+\Delta z)+h(x,y,z-\Delta z)-2h(x,y,z)}{(\Delta z)^2}\right) \quad (21)$$

The partial derivative of *f* with respect to eta is also calculated for the discretization process. Written out, the equation for *f* is given as Eq. 22.

$$f(\rho,\eta_0,\eta_1,\dots,\eta_N) = K_1\left(\rho^4 + \frac{-4\rho_{vap}-4\rho_{sol}-2}{3}\rho^3 + \frac{4\rho_{vap}\rho_{sol}+2\rho_{vap}+2\rho_{sol}}{2}\rho^2 - 2\rho_{vap}\rho_{sol}\rho\right) + K_2\Big(\rho^2 + 6(1-\rho)(\eta_0{}^2+\eta_1{}^2+\cdots+\eta_N{}^2) - 4(2-\rho)(\eta_0{}^3+\eta_1{}^3+\cdots+\eta_N{}^3) + 3(\eta_0{}^4+\eta_1{}^4 + \cdots+\eta_N{}^4) + w(\eta_0{}^2\eta_1{}^2+\eta_0{}^2\eta_2{}^2+\eta_0{}^2\eta_3{}^2+\cdots+\eta_0{}^2\eta_N{}^2+\eta_1{}^2\eta_0{}^2+\eta_1{}^2\eta_2{}^2+\eta_1{}^2\eta_3{}^2+\cdots + \eta_1{}^2\eta_N{}^2+\cdots+\eta_N{}^2\eta_0{}^2+\eta_N{}^2\eta_1{}^2+\eta_N{}^2\eta_2{}^2+\cdots+\eta_N{}^2\eta_{N-1}{}^2)\Big) \quad (22)$$

The partial derivative of *f* with respect to η_i for all *i* is given as in Eq. 23.

$$\frac{\partial f}{\partial \eta_i} = 12K_2\left((1-\rho)\eta_i - (2-\rho)\eta_i{}^2 + \eta_i{}^3 + \frac{w}{3}\eta_i\sum_{j\neq i}^{N}\eta_j{}^2\right) \quad (23)$$

Using the forward euler method for stepping forward in time, *f* at the next time point (t+Δt) is given in Eq. 24.

$$f(t+\Delta t) = f(t) + \frac{\partial f}{\partial t}\Delta t \quad (24)$$

The spatial domain is discretized with pixels so that for a pixel at locations (a,b,c) the next pixel in the positive x direction is (a+1,b,c). So, $\Delta x = \Delta y = \Delta z = 1$. Using the above equation (Eq. 24) for marching forward in time and also noting the gradient in x,y,z is 1 pixel, the equations for ρ and η at the following time step is given by the Eq. 25 and 26.

$$\begin{aligned}\rho(t+\Delta t) &= \rho(t) + \frac{\partial \rho}{\partial t}\Delta t = \rho(t) + (\nabla D \nabla h + D\nabla^2 h)\Delta t \\ &= \rho(t) \\ &+ \Delta t[D(x+\Delta x,y,z)[h(x+\Delta x,y,z)-h] - D(x-\Delta x,y,z)[h-h(x-\Delta x,y,z)] \\ &+ D(x,y+\Delta y,z)[h(x,y+\Delta y,z)-h] - D(x,y-\Delta y,z)[h-h(x,y-\Delta y,z)] \\ &+ D(x,y,z+\Delta z)[h(x,y,z+\Delta z)-h] - D(x,y,z-\Delta z)[h-h(x,y,z-\Delta z)] \\ &+ D(h(x+\Delta x,y,z)+h(x-\Delta x,y,z)+h(x,y+\Delta y,z)+h(x,y-\Delta y,z)+h(x,y,z+\Delta z) \\ &+ h(x,y,z-\Delta z)-6h)] \end{aligned} \quad (25)$$

$$\begin{aligned}\eta_i(t+\Delta t) &= \eta_i(t) - L\left(\frac{\partial f}{\partial \eta_i} - \beta_\eta \nabla^2 \eta_i\right)\Delta t \\ &= \eta_i(t) \\ &- L\Delta t\left(12K_2\left((1-\rho)\eta_i - (2-\rho){\eta_i}^2 + {\eta_i}^3 + \frac{w}{3}\eta_i\sum_{j\neq i}^{N}{\eta_j}^2\right)\right. \\ &- \beta_\eta(\eta_i(x+\Delta x,y,z)+\eta_i(x-\Delta x,y,z)+\eta_i(x,y+\Delta y,z)+\eta_i(x,y-\Delta y,z)+\eta_i(x,y,z+\Delta z) \\ &\left.+\eta_i(x,y,z-\Delta z)-6\eta_i)\right) \end{aligned} \quad (26)$$

Eqs. 25 and 26 are the final discretized equations used in the simulations. The equations were translated into a simulation using programming in C++.

2.2 PFM Simulations

In this section the setup of the PFM simulations is discussed. First the need to transition from 2D to 3D simulations is explained in more detail, as well as the complexities involved in making this transition. Next the parallelization process is discussed, and the simulation parameters are listed and explained. Then the steps for running the simulations are discussed. The bed generation tool used for generating the beds is presented and then

the python post processing algorithms are explained. Finally, a summary of the simulation procedure is outlined.

2.2.1 2D TO 3D

As discussed earlier, previous simulations have been done on a number of particles in 2D. However, these 2D simulations are unable to account for the volumetric changes taking place in the simulation. For example, these simulations are unable to account for z shrinkage in the bed or changes in relative density with sintering. As such, the simulation used in this project was built first for beds in 2D and then transitioned to 3D simulations. With scaling the simulation from 2D to 3D an additional dimension is added and the number of particles are increased. So, the size of the simulation is scaled up by millions of pixels. This simulation size takes a lot of time to run on a single computer unit so a Message Passing Interface (MPI) was used to reduce the amount of time that the simulation takes to run.

2.2.2 PARALLELIZATION

A Message Passage Interface (MPI) was used to break the simulation box up into smaller boxes. The use of MPI allows the processors to communicate and share information among the edges of the box boundaries. Initially the simulation was broken up only along the z axis. This analysis simplified the process of transferring information amongst the processors as only the neighboring z information is transferred, and it is simply a matter of concurrent numbers to determine what processors should be sending what information to which processors. However, the major drawback with this is that there is a restriction on the amount of parallelization that can be done, which means that although the simulation

takes less time than a single unit, it still takes a lot of time to run as parallelization is only done along one axis.

Next the simulation was parallelized in both the z and y axis. With this the simulation can be broken down along an additional axis which further reduces the amount of computational time. This process increased the complexity of aligning the simulation accurately in terms of lining up the accurate boxes broken along the y and z axis in order to get an accurate picture of the simulation bed. This problem expands to deciphering the accurate processors to send information to and the corresponding processors to receive it. For the first and second order differentiation each smaller box required the neighboring z and y axis edge bounds and the second edge bound. The breaking of the simulation box and the required information to send is illustrated in Figure 2.1.

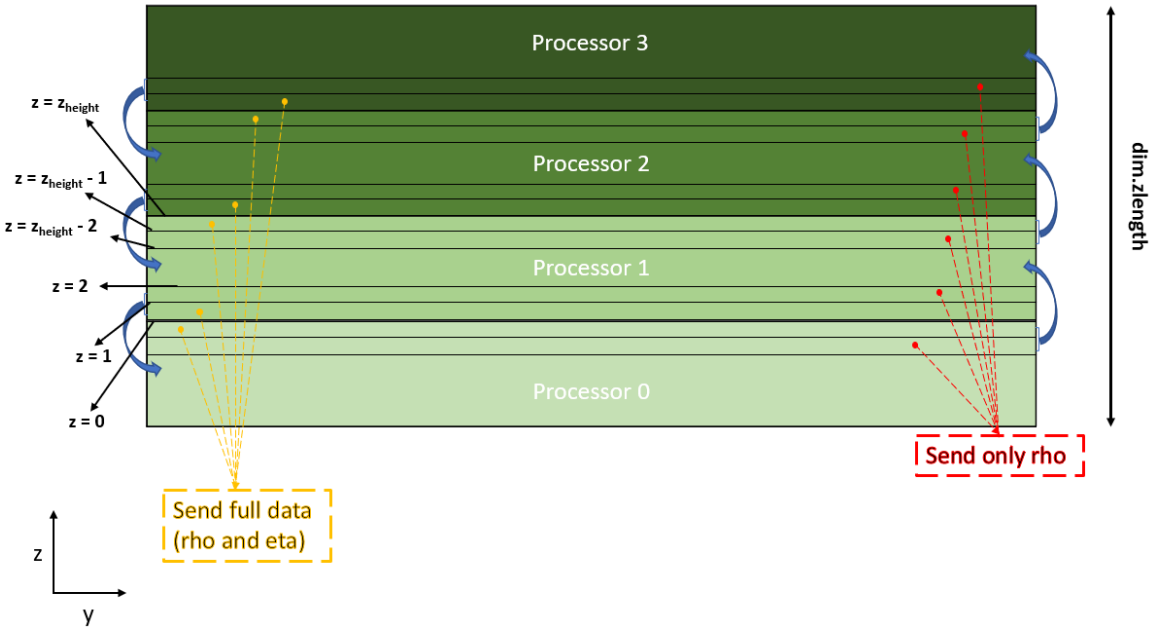


Figure 2.1a: Parallelization of simulation box in z

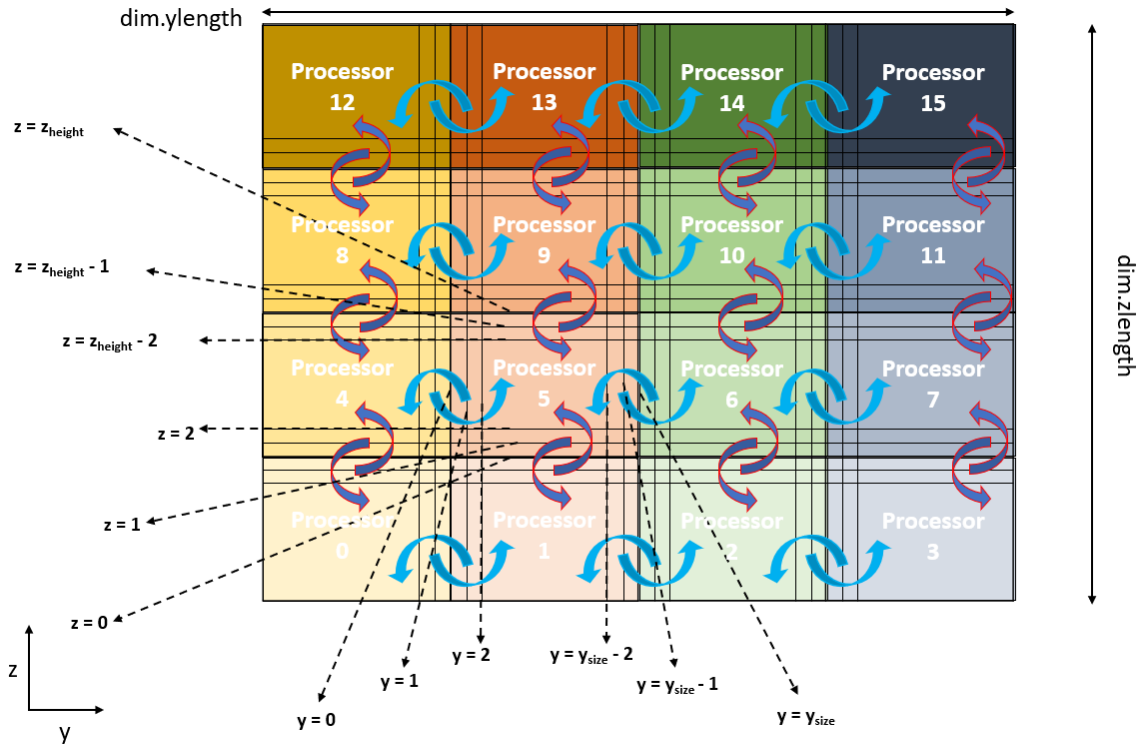


Figure 2.1b: Parallelization of simulation box in y and z

Where z_{height} and y_{size} are the full z and y sizes of the boxes given to each processor. In general they are defined as the total length (dim.zlength and dim.ylength) divided by the total number of processors allocated for z and y. These constants are defined in more detail in the following section.

2.2.3 SIMULATION PARAMETERS

The simulation set up parameters are listed in Table 2.1 as well as their corresponding functions.

Simulation parameters	Function
t	
con.deltaT	
finalt	Timing control
N_size	Neigbouring cells (set as 6 for 3D)
dim.xlength	
dim.ylength	
dim.zlength	
dim.noOfparticles	
psizey	
psizez	Constants related to box sizes
intvs	
intvf	Control printing output of results
startruns	
runno	
noruns	
maxruns	
fileno	
progno	
simno	Related to automatically varying the sintering parameters
pid	Master processor (set as 0)
limR	
cutoffR	Control conservation of density variable
dim.etaSolid	
dim.etaVap	
dim.rhoSolid	
dim.rhoVap	
con.randE	
con.randR	PFM constants
con.gbmobility	
con.vapdif	
con.A	
con. B	
con.Cgbe	
con.gbenergy	
con.surfenergy	
con.gbdif	
con.surfdif	
con.voldif	Sintering parameters
SNo	Sintering parameters tag

Table 2.1: Simulation setup parameters

N_size in Table 2.1 is a constant which controls the arrays for the neighboring cells. This constant is set based off the number of neighboring cells of each pixel. For a system in 3D, *N_size* is fixed at 6 (for 2D *N_size* would be 4). The variables related to varying the sintering parameters automatically are only used in the case when the simulation is used in exploration mode.

Exploration mode in this simulation is used to study the effect of varying sintering parameters on the rate of densification. When this mode is used, the sintering parameters to be varied are set in a constants file set by the *fileno* variable. The *runno, startruns*, and *maxruns* variable are used to set the specific line in the file to read (if one line) or the range of lines. This mode may be used for more than just checking densification rates, and it may also be used to test if the simulation remains stable for a set of constants. This is the purpose of the *progno* and *simno* variables. The *progno* variable defines the output progress file which reports the stability of a set of sintering parameters defined in the input file. The *simno* variable is a tag specific to each sintering parameter defined in the input and output progress file. The form of the input and progress files are: “infoconst<fileno>.txt” and “Progress_file<progno>.txt” respectively.

The *pid* variable sets the value of the master processor. This master processor is assigned the tasks of relaying information back to the user. The *pid* value should not be higher than the total number of processors used in the system. Setting *pid* as 0 or 1 is safe, as there is rarely ever less than two processors used. The default values of the PFM constants used in this simulation are listed in Table 2.2.

PFM constants	Default value
dim.etaSolid	1
dim.etaVap	0
dim.rhoSolid	0.9998
dim.rhoVap	0.000000089
con.randE	0
con.randR	0.0000001

Table 2.2: PFM initiation constants

Table 2.2 shows the constants used to initiate the simulation box. *Dim.etaSolid* and *dim.etaVap* set the order parameter for the ith particle. When the order parameter is in a particle it takes on the value of *dim.etaSolid*, in another particle it takes the value of *dim.etaVap*. Similarly, the *dim.rhoSolid* and *dim.rhoVap* values differentiate between the solid and vapor phase. The value for the density in vapor phase was taken to be 0.000000089 and in solid phase 0.9998. These values were chosen instead of the standard values of 0 and 1, to allow for numerical analysis of the discretized differential equations. When a particle is solid the density variable takes on a value of *dim.rhoSolid*, in vapor it's *dim.etaVap*. *con.randE* and *con.randR* are used to randomize the array of initial values chosen. The other functions are discussed in more detail in the following subsections.

2.2.3.1 Simulation box sizes

There is a restriction with the number of processors used to break the simulation box up in the y and z axis. This restriction is that the sizes in the y and z axis must be directly divisible by the number of processors assigned in the y and z. This restriction exists to ensure that when the simulation box is broken up amongst the processors assigned, the full range of the simulation box is covered. If this isn't done correctly, parts of the

simulation box may be unaccounted for. Table 2.3 has one example of parameters used to control the simulation box size.

Constants related to box size	Example values
dim.xlength	67
dim.ylength	128
dim.zlength	116
dim.noOfparticles	33
psizey	16
Psizez	4

Table 2.3: Constants related box size

For the example in Table 2.3 the total number of processors is 64. The *dim.xlength* variable is actually the height of the simulation box. The x and z values are switched because the x axis is not parallelized, so the most efficient use of resources would be to use the smallest box size length in this axis.

2.2.3.2 Printing results

Table 2.4 shows the parameters controlling file output and the file format name associated with the variables. *Intvs* and *intvf* parameters in the table control the interval of outputting the results. In the example in the table below, the values correspond to the simulation time variable and not the number of simulation time steps.

Printing parameters	Output file format	Example
intvs	3p and rho	0.5
intvf	full	20

Table 2.4: File output control

The results are output in three main formats as seen in the table: 3p, rho and full. Data printed in 3p is the sum of the values of all the eta variables multiplied by the rho variable for each pixel. Printing out the information in this form allows for better visualization of the data because of the inclusion of the eta variables. It doesn't require as much data as the full so more data time points can be printed out with this option. Data printed with the rho option prints out only the rho variable for each pixel. This option is required for post processing analysis done on the beds. The rho option is crucial as it gives the results for the conserved variable and is why the analysis can be done on the data output. Finally, the full option prints out all the variables separately. It prints the rho and eta variables for each pixel. The eta variable corresponds to the number of particles in the system so for 35 particles there are 35 different eta variables and 1 rho variable. To completely quantify this, for a 100 by 100 by 50 pixel box, the 3p and rho options will give outputs with a file containing 500,000 points. On the other hand, the full option would give an output containing 18,000,000 points (printing out rho and eta for each pixel). The full option is required to start the simulation at a time point that is not 0. So, to allow the simulation to start at a later time, all the information present in the simulation is required. However, the large file size with the full option makes it infeasible for the full data to be printed out with as high a frequency as the other options. The name of the output file is set as shown in the figure below.

```
MPIfileprintsum(t,evat,con.deltaT,Eta,"3p","0.0001.dat",100*evat,procSize,prank,dim,zpos,ypos,zheight,ysize);
MPIfileprintrho(t,evat,con.deltaT,Eta,"rho",".dat",100*evat,SNo,procSize,prank,dim,zpos,ypos,zheight,ysize);
MPIprintfullE(t,evatf,con.deltaT,Eta,"fullT",".dat",10*evatf,SNo,procSize,prank,dim,zpos,ypos,zheight,ysize);
```

Figure 2.2: Controlling output filename

The blue words in Figure 2.2 are the functions called for printing out the sum (3p), rho and full information. The yellow words set the name of the output file and the white lines are parameters which go into the output file functions and are used to set the different bounds to print for each processor and give additional information to ensure that input files are unique with time and sintering parameter sets.

2.2.3.3 Start time and simulation input format

The first constant in the simulation script (*ffile*) differentiates between starting the simulation at the beginning 0, or a later set time. If the simulation starts at 0 the input file is a text file which contains the radii of the particles as well as their x, y, z coordinate centers in pixels. Figure 2.3 shows the input format of the desired file and the naming of the file in both cases where simulation starts at 0 and another time.

```
ifstream particleInfofile ("particledata43.txt");

for (int i = 1; i <= dim.noOfparticles; i++)
{
int j = i-1;

particleInfofile>>circleProp[j][0]>>circleProp[j][3]>>circleProp[j][1]>>circleProp[j][2];
}
```

filename

Figure 2.3a: Reading information from input file when simulation starts at 0

Figure 2.3b: Reading information from input file when simulation does not start at 0

Figure 2.3 shows the setup location of these files in the C++ simulation script. In Figure 2.3a, *circleProp* sets the starting information used to separate the pixels in the simulation box into different spherical particles. Here *circleProp[j][0]* represents the radius, *circleProp[j][1]* represents the x, *circleProp[j][2]* represents the y and *circleProp[j][3]* represents the z of the center of the particles. The order the numbers are placed in correspond to the order to which the data should be read in. If the file is to be started at a later set time, the input file must be set. The input file corresponds to the naming of the full file which contains all the information for each pixel in the box. With this the assumption is made that the input file corresponds exactly to the set simulation size. An additional test which could be done to ensure this is to visualize the results running through the simulation to make sure it starts where the user wishes. Also, if continuing, the time in the loop where the input file is set (labeled Simulation start time in Figure 2.3b) has to be set to the desired start time.

2.2.3.4 Conservation of density variable

Initially running the simulation using only the equations listed above did not enforce conservation of the density variable. Additional conditions had to be included to enforce conservation of density. This is done using a scheme which makes use of two constants. The first constant is the cut off constant which separates pixels in the vapor phase and the solid phase. Pixels with rho variables having values less than the cut off are taking to be in vapor phase. The limit value determines the substantial variables. The pixels with density variables greater than the limit value are used to count the number of substantial solid variables. The difference between the initial sum of the density variable and the sum at each step is divided by the number of substantial pixels and is added back into the system amongst the substantial pixels. Default values for these constants are listed in Table 2.5.

Density conservation	Defaults
limR	0.0041
cutoffR	0.0001

Table 2.5: Default density conservation variables

2.2.3.5 Sintering parameters

The term sintering parameters and sintering constants have been used interchangeably in this thesis because the sintering parameters are constant with respect to time but not temperature. The parameters in Table 2.6 correspond to the constants in equations 1 to 26. *Con.gbmobility* corresponds to L in Eq. 7, *con.surfdif*, *con.gbdif*, *con.voldif* and *con.vapdif* correspond to the diffusion coefficients in Eq. 5. *Con.surfenergy* and *con.gbenergy* correspond to the energy gradients β_ρ and β_η in Eq. 1. *Con.A*, *con.B* and *con.Cgbe* correspond to K_1, K_2, and w in Eq. 2. *SNo* in the table is the tag which corresponds

to this set of sintering parameter values. The constants in Table 2.6 correspond to arbitrarily chosen constants for the first default run of the simulation. These constants were arbitrarily chosen to correspond to a 10:1 ratio of surface to grain boundary diffusion and a 2:1 ratio of surface to grain boundary energy, derived from Eqs 3 and 4.

Sintering parameters	Default values
con.gbmobility	10
con.vapdif	0
con.A	12
con. B	1
con.Cgbe	7
con.gbenergy	3.75
con.surfenergy	10
con.gbdif	4.1
con.surfdif	41
con.voldif	0.041
SNo	1

Table 2.6: Default sintering parameter values

2.2.4 RUNNING SIMULATION

In this section the details of running the simulation in the Texas Advanced Computing Center (TACC) system is presented as well as some of the experienced outputs. TACC is a supercomputing environment consisting of thousands of nodes which makes fast and large-scale computing possible.

2.2.4.1 Job script

The simulation is run with a job script. This job script is scheduled through a SLURM scheduler [28]. An image of a typical job script is included in Figure 2.4.

```
#!/bin/bash

#SBATCH -J bed          # Job name
#SBATCH -o bed.%j.out    # stdout; %j expands to jobid
#SBATCH -e bed.%j.err    # stderr; skip to combine stdout and
stderr
#SBATCH -p normal     # queue
#SBATCH -N 1              # Number of nodes, not cores (64
cores/node)
#SBATCH -A Modeling-of-Microsca         # Project Allocation
#SBATCH -n 64             # Total number of MPI tasks (if
omitted, n=N)
#SBATCH -t 48:00:00       # max time

#SBATCH --mail-user=<email address>@utexas.edu
#SBATCH --mail-type=ALL


ibrun ./bed1
```

Figure 2.4: Job script for submitting and running simulation on stampede2

The script is submitted via the stampede2 terminal command: ‘sbatch <job script name>’. Stampede2 was chosen over the other TACC supercomputers because it has the fastest performance for C++ applications. More information about using stampede2 can be found in the user guide [29]. As discussed, the total number of processors requested has to be equal to the multiple of the y and z processors set in the code (section 2.2.3.1). The figure shows the explanations line by line for each command in the job script. The maximum time allocated for the run of this simulation depends on the compute nodes used. In the script in Figure 2.4, the KNL compute nodes are used and a typical run time for a one-by-one micrometer bed is 36 hours. This time is shorter when using the SKX nodes (these nodes have a different number of processors in each core so care must be taken in toggling

between these compute nodes). The last line is the command to run the executable of the C++ simulation script. This executable is compiled through the terminal.

2.2.4.2 Experienced Outputs

There are three types of outputs which have been seen commonly from this simulation. Each of these outputs are described and shown in the figures in this section (Figures 2.5 – 2.7). Figures 2.5b, 2.6a and 2.7b show three columns. The first gives the sum of density at the initial time step. The second shows the sum of the density variable at the current time step before the density conversation scheme is enforced. The last column shows the number of pixels with density variables above the limit constant set. Between each row the simulation timestep is shown.

2.2.4.2.1 Successful

Figure 2.5 shows the expected output when the simulation is successfully completed.

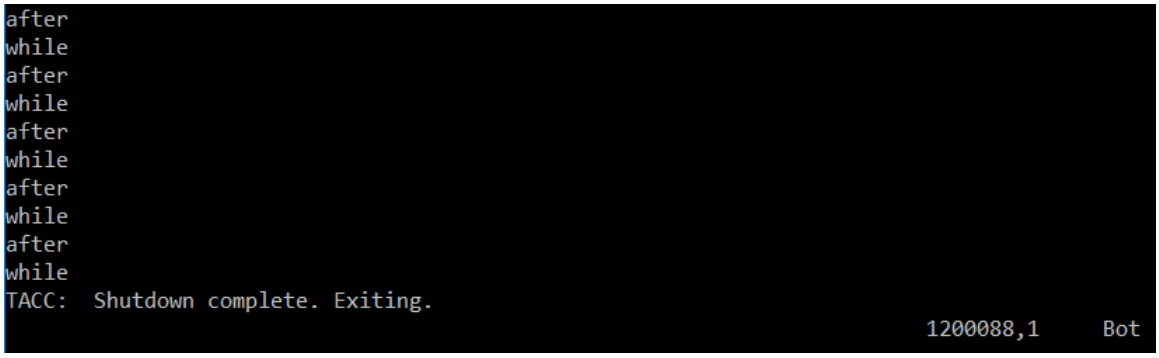


Figure 2.5a: Bottom of successful output file

Figure 2.5b: Successful output file: Last printed result before exit

In this case the output file shows that the simulation has been completed and all data allocations have been freed. The ‘after while’ displayed in Figure 2.5a is the completion message printed by all processors after successfully completing the iteration loop. In a successful run like Figure 2.5b, the file runs all the way to the final time and then the after while messages are displayed.

2.2.4.2.2 Unsuccessful due to incomplete time

The job script allots the amount of time the user deems fit to run the simulation to the prescribed end. If, however, this time is not sufficient to see the simulation to the specified end, the output file ends without the successfully completed message shown in the image in Figure 2.5. In this case an additional message is sent to the email specified in the job script. An example is shown in Figure 2.6b.

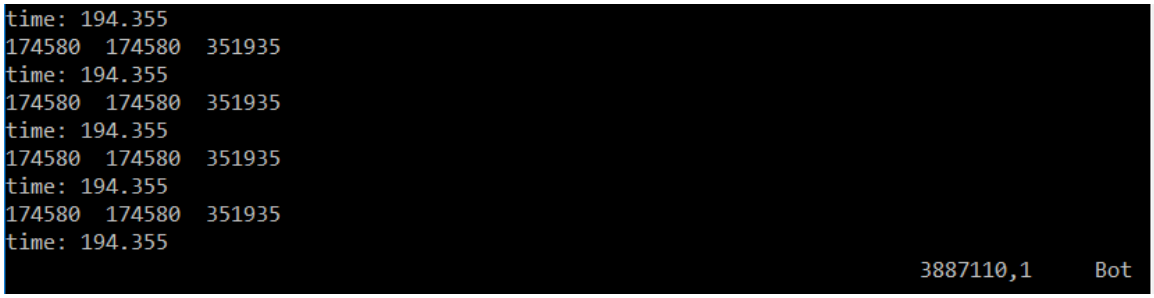


Figure 2.6a: Bottom of time insufficient output file

Figure 2.6b: Insufficient time: email title sent to specified address

Figure 2.6a shows that when the simulation is prematurely ended due to insufficient time, the simulation does not run to the specific final time and no 'after while' message is printed, showing that the simulation ended with the processors still in the while loop doing time step iterations.

2.2.4.2.3 Instability

To prevent wasting computing allocation, a failsafe is added to check for the stability of the simulation. This is done by adding the density variables together and checking to see if the total value in any processor is greater than 10 times the initial sum of the density variable. The ratio 10 was chosen as a maximum bound to capture instability (values lower can this can also be used). If any processor satisfies this condition the simulation ends, and the output message seen corresponds to the image in Figure 2.7b.

```
while
after
while
TACC:  Shutdown complete. Exiting.
                                                                    51345,1       Bot
```

Figure 2.7a: Bottom of unstable output file

```
time: 2.5602
166793  166842  338965
time: 2.5603
166793  2.85586e+06     338947
time: 2.5604
166793  3.12512e+48     148085
time: 240
-->Simulation unstable: Prematurely ended!<--
time: 240
after
                                                                    51170,1       99%
```

Figure 2.7b: Unstable output file: Last printed result before exit

As seen in Figure 2.7a even though the simulation stops prematurely the ‘after while” output shows that the simulation is ended successfully, which means that the processors close down and the scheduled job ends.

2.2.4.2.4 Miscellaneous

One other reason the simulation could fail is the faulty allocation of processors. That is to say that if the processors in the batch script is less than the allocation in the simulation code. Another reason may include insufficient storage space for the files allocated in the code. Also, if there is insufficient computer allocation for the required simulation then it will result in an incomplete simulation run.

2.3 Bed generation

The simulations used in this project are tested first with 2D systems and a few 3D particles and then it is transitioned to particle beds. These particle beds are created using a

bed generation tool. The bed generation tool used generates particles arranged to match an actual physical bed. These beds consist of spherical nanoparticles, generated by setting an initial position vector for each particle as well as a radius, to set particle size. The generation simulation uses Discrete Element Modelling (DEM), in a Multiphase Computational Fluid Dynamics simulation, MFIX. Particle packings are generated using the MFIX-DEM discrete mass inlet function, which allows each particle to interact with neighboring particles. Initially, the particles are distributed randomly within the bed domain, they are given an initial velocity and are allowed to move within an initial set of boundary conditions. Particle interactions are modeled using a dashpot model based on a soft-sphere model of the particles. Subject to gravitational and cohesive forces, the particles move around and interact until the final steady state position is reached. This final geometry is used as the input into the PFM simulation. Complete details of the bed generation process can be found in reference [30].

2.4 Post processing with python

Post processing is required to take in the data output from the C++ simulation, which gives the values of the density and eta variables, and translate it into data which can be used for further analysis of the sintering process. This section contains the details of the visualization and data analysis scripts used for running the simulations.

2.4.1 VISUALIZATION

For 3D visualization the pixels are drawn as circles with different border colors than that used for the fill in the center of the circle. Some inputs to this script include the size of the simulation box. This should correspond to the simulation sizes used in the C++ code described in the PFM simulation section 2.2. Another input is the file type. This

should match the output from the sintering simulation (Printing results section 2.2.3.2). The python algorithm is made specifically for the output for the 3p and rho file output formats. It does not work for the full data output. The range of data files to plot is an added input. This includes the start file, which is the first data file to be visualized, the end file, last file to be visualized, and the interval to go between in plotting the data in these files. This interval can be the interval the files were created in in the sintering simulation, or a larger interval if some files should be ignored. Also, the form of the output image files should be stated. In the case where the parallel option is used, an additional input to the simulation is the number of processors desired for parallelization, this lets the program know how many processors to divide the work between. The number of processors listed should not be above the number available in the system. With this option an additional output is the total number of files to be analyzed (corresponds to number of sets to run in a single thread simulation) and the number of sets to be run when distributed amongst the processors. This number should be significantly lower, depending on the number of processors, as there are now a number of processors analyzing different files at the same time.

2.4.2 DATA ANALYSIS

The data analysis algorithm works with the same inputs as described for the visualization algorithm. Data analysis is done on the center analysis box in the middle of the simulation. This is to make sure that the data collated is impervious to edge effects. In this analysis box, the calculations for density were found by taking the ratio of the total sum of the conserved density variable in the box, to the total volume of that box. With this method, the relative density has a value of 0 in a fully porous box and a value of 1 when

fully dense. To get an uncertainty measure for the analysis done on these beds, several boxes were taken from each bed. These boxes were chosen based off an edge finding algorithm.

2.4.3 EDGE FINDING ALGORITHM

The purpose of the edge finding algorithm is to determine the largest possible bounds for analysis without encroaching upon the edges of the simulation. The significance of this tool is the ability to determine the cutoff point between the fully populated bulk of the bed and the edges of the simulation which undergo x-y shrinkage. This algorithm assumes that there is a larger deviation of the bottom of the bed from the base of the simulation box the closer one gets to the edges. This is evident in the rounding at the edges shown in the example slices in Figure 2.8.

The first version of the edge finding algorithm used slices in the x and y direction to calculate the deviation from the top and base of the bed. An example is shown in Figure 2.8.

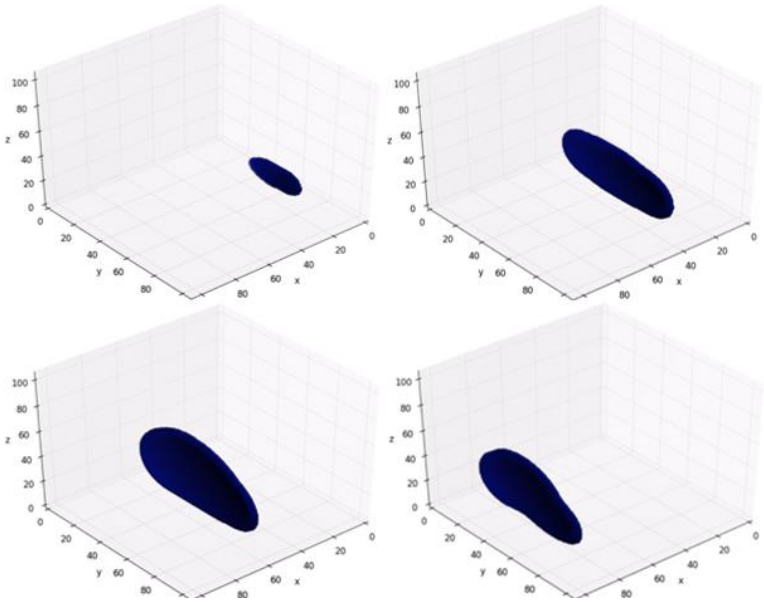


Figure 2.8: Example of x slices taken at different locations in the bed

In each slice the deviation from the highest and lowest regions of the bed is calculated, and the total deviation is defined as the Euclidean norm of these values. The total deviation is then normalized against the largest difference in deviations for each bed. This gives the τ value found for the algorithm and is shown in Eq. 27 below. This method was found to introduce accumulated errors, combining valid regions in the middle of the bed with invalid regions at the end. This problem was corrected with the updated version.

$$\tau_i = \frac{\sqrt{\left({t_{D,i}}^2 + {b_{D,i}}^2\right)}}{max\left[\sqrt{\left({t_D}^2 + {b_D}^2\right)}\right] - min\left[\sqrt{\left({t_D}^2 + {b_D}^2\right)}\right]} \tag{27}$$

The updated version of the edge finding algorithm works by taking strips of the full z height, as shown in Figure 2.9.

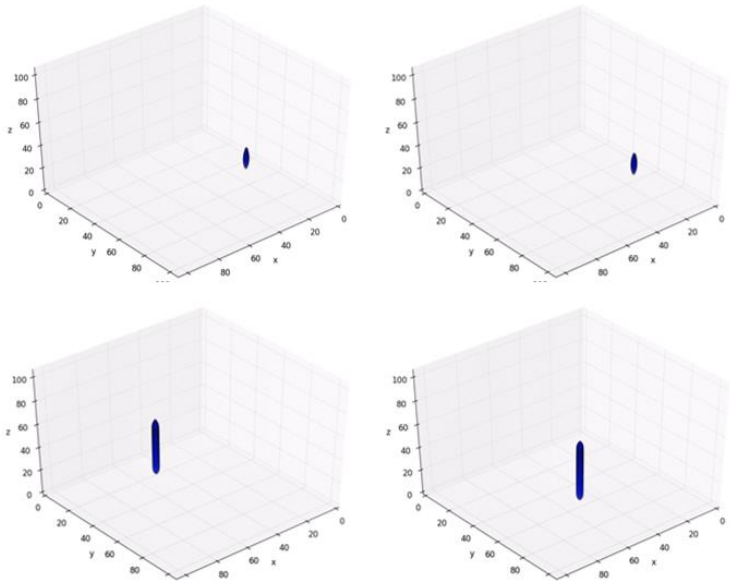


Figure 2.9: Example of z strips taken at different locations in the bed

In each z-strip the deviation from the bottom of the bed to the base of the simulation box is calculated and normalized against the largest difference in a z strip throughout the bed. This gives the τ value found for the algorithm and is shown in Eq. 28 below. The τ value takes on a value of 1 when a z-strip is completely empty, a value of 0 when the strip has no deviation from the bottom, which implies that the strip is flat and is touching the base of the simulation box, and values between 0 and 1 at intermediate stages.

$$\tau_i = \frac{t_{D,i}}{max[t_D] - min[t_D]} \tag{28}$$

where t_D refers to the deviation of the z strip from the base of the simulation box. Once τ is calculated, the algorithm compares this value to a predetermined cut off factor and defines the analysis bounds as the x and y values that give τ values just below the cut off. The optimum cut off factor for the simulation was determined analytically by varying different factor values to get the best agreement for all beds tested.

2.4.4 RELATIVE DENSITY

After determining the appropriate bounds for the simulation, the relative density is calculated from 40 by 40 pixel boxes chosen from within the pre-determined acceptable bounds. The relative density is defined in Eq. 29.

$$\rho_{rel} = \frac{\rho_i - \rho_o}{\rho_i} \tag{29}$$

where ρ_o is the sum of the density variable at the start of the simulation, and ρ_i is the sum at the simulation timestep i.

2.5 Summary

The overall flow of the simulation procedure is shown in Figure 2.10 below.

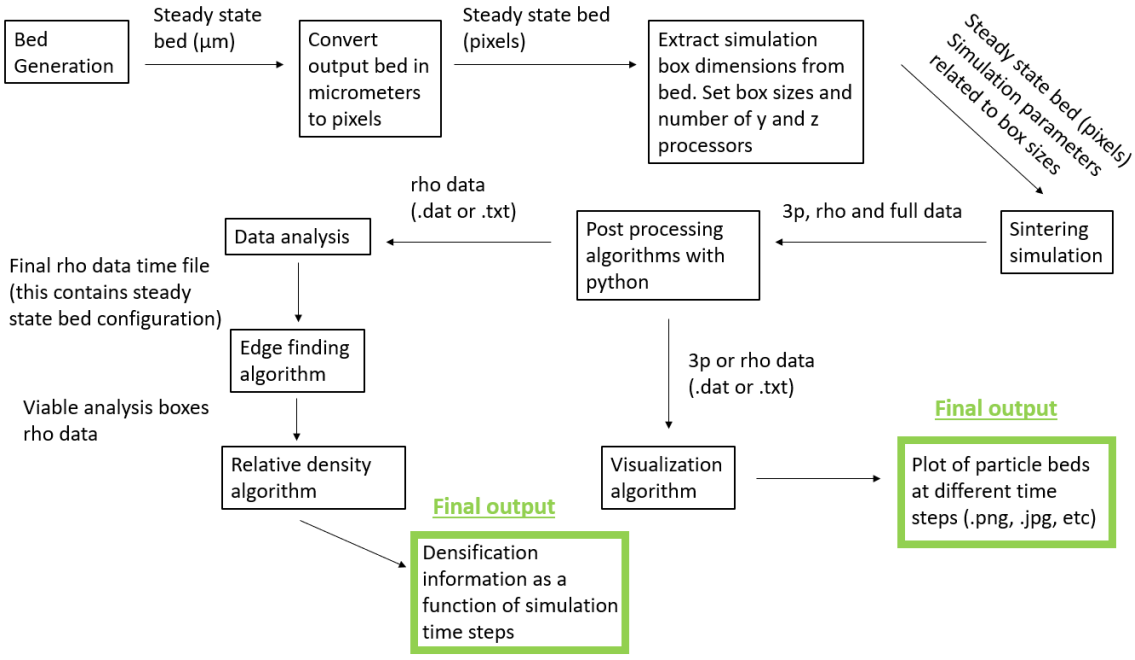


Figure 2.10: Flow of simulation procedure

The chart in Figure 2.10 shows the flow of simulation procedure starting from the first stage of bed generation, all the way down to the final outputs from the simulation. The information written above the arrows are the inputs into each step and outputs from the previous step. The 3p, rho and full data outputs from the simulation can either be printed out into a text file or a binary data file. The binary data file stores the most complete data but are not directly human readable. The text files lose float information, but they are human readable. Either of these formats can be used for the post processing algorithms. Whichever format is used must be set in the python scripts.

After the development of the simulations were concluded, the experimentation phase began. The experimental procedure is discussed in the next chapter.

CHAPTER 3: EXPERIMENTAL PROCEDURE[2]

Experiments were performed to validate the results of the PFM simulations and to calibrate the sintering parameters to sintering temperature. This chapter discusses the flow of the experiments. First the sample is prepared, then the sintering is performed on the samples and then density measurements are derived from the experiments. These processes are discussed in more detail in the following sections.

3.1 Sample preparation

Sintering experiments were carried out on copper nanoparticle inks from Intrinsiq [31]. The first step of the experimental procedure involved drying out the ink. 2ml of copper nanoparticle ink was dispensed into a glass petri dish using a rubber pipette. The petri dish of ink was dried on a hot plate at around 95˚C for 16 hours. After the solvent in the ink dried off, dried copper flakes of coated nanoparticles were scraped off the petri dish with a flat spatula. These dried flakes were then put into crucibles. Pressure was applied to form the flakes into pellets in the crucibles. These crucibles were subject to isothermal heating. During the process of heating, the coating around the particles degraded and the nanoparticles sintered together into a solid pellet. The flow of the experimental procedure is shown in the images in Figure 3.1.

[2] Dibua OG, Yuksel A, Roy NK, Foong CS, Cullinan M. Nanoparticle Sintering Model: Simulation and Calibration Against Experimental Data. ASME. J. Micro Nano-Manuf. 2018;6(4):041004-041004-9. doi:10.1115/1.4041668.

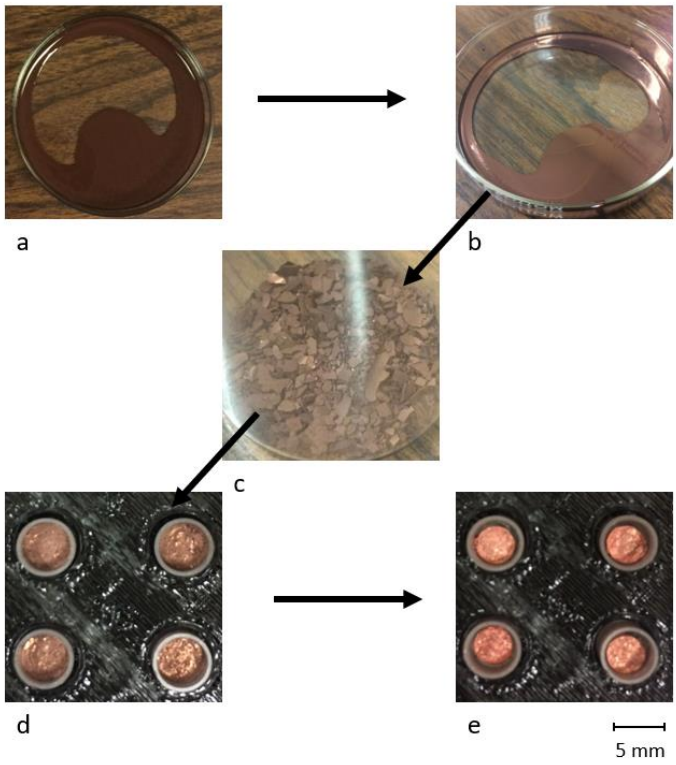


Figure 3.1: Experimental procedure. a. Copper nanoparticle ink. b. Dried ink. c. Scraped off dried flakes. d. Pellets in crucible before sintering. e. Pellets in crucible after sintering

3.2 Heating experiment methods

The experiments were carried out using two processes the first is a Thermogravimetric Analysis (TGA) and the second is furnace heating. For both of these processes the samples were prepared the same way. The different sintering methods are discussed in this section.

3.2.1 THERMOGRAVIMETRIC ANALYSIS (TGA) HEATING

The heating curve of the TGA experiments are shown in Figure 3.2.

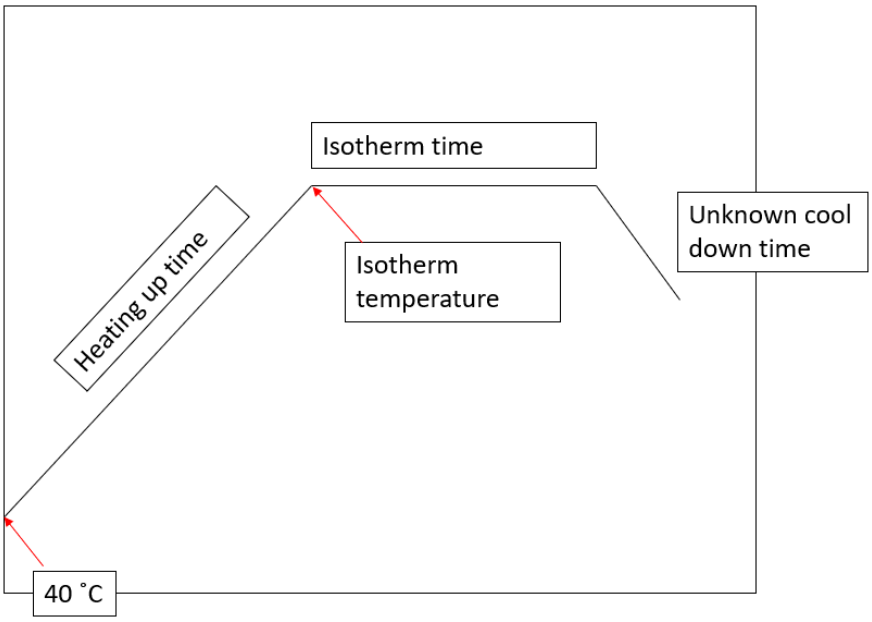


Figure 3.2: Heating cycle of TGA experiment

The set up for the sintering experiments using this device was easy as the TGA includes an automatic loading and unloading system. As such, the samples are put into slots in the waiting area and the heating cycle is set. For these experiments this cycle involved an initial period of gradient heating till the system and the sample reached the desired temperature and a period of isothermal heating at the desired temperature for a set time. With this equipment the desired experiments are programmed in and the samples are loaded into the heating area after the previous sample is done heating. There is very little human effort involved in loading the sample. The experiments were carried out under flowing 50ml of nitrogen.

3.2.2 FURNACE HEATING

The furnace used for these experiments is made up of a heater, a quartz glass tube, and valves for flowing reducing gasses. To perform the experiment in this setting a crucible

holder is required to control moving the sample into and out of the heated area in the quartz tube. Images of the sample holder and furnace set up are shown in Figure 3.3.

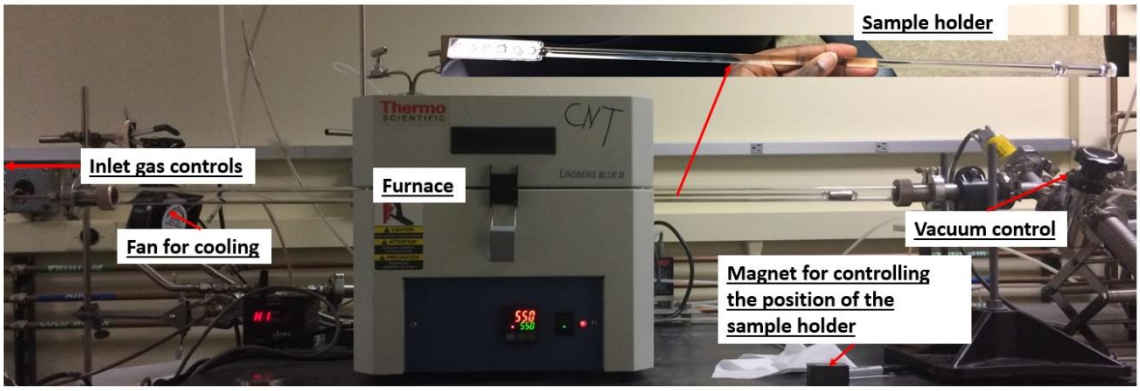


Figure 3.3: Furnace heating setup

During the experiment, the sample crucible is put into the glass holder in the tube and the tube is set up and secured to ensure that there is no gas leakage and the environment is properly purged. Once the right gas flow is established, the crucible in the glass holder is moved into the heated region using a magnet. To allow for this carriage the head of the glass holder has a slot which allows for a thin strip of magnetic metal to be wrapped around the holder allowing for transportation in the tube. Once the temperature is set in the oven the sample is moved in, left in for the amount of heating time desired, and taken out, then the oven is set to cool and this process is repeated for as many temperature and time cycles as is required. The process of moving the holder into and out of the heated zone is done manually. The experiments took place under flowing argon and hydrogen in the ratio Ar : H 300:100.

3.3 Measurements

Analysis of the sintered copper pellets were carried out after the experiments. The density measurement is calculated as the ratio of mass to volume. The mass was measured using a digital weight scale and the height of the copper nanoparticle pellet in the crucible was measured using a plastic stopper. First, the height of the pellet was calculated as the difference between the height of the stopper when placed in the empty crucible, and the height when placed in the crucible containing the pellet. The volume was then calculated using this measured height and the diameter of the crucible, for density measurements before sintering, or the diameter of the pellet formed, for measurements after sintering. Similarly, the mass of the pellet was calculated as the difference between the mass of the empty crucible and the mass of the crucible containing the pellet. The initial density of the pellet was calculated from the ratio of mass to volume. The mass measurement used in these calculations is the mass measured after sintering. The assumption used here is that prior to sintering the sample contains the nanoparticle coating as well as any undried solvent. As such, the true mass (more accurate mass) is that measured after sintering after the polymer coating decays off. The relative density was then calculated based off the measurements taken. This value was calculated using the same equation as was used for the simulation (Eq. 29), where in this case ρ_o corresponds to the initial density of the unsintered pellet and ρ_i is the density after sintering. These values are then fit to an analytical equation for calibration against simulation data.

After the experiments were performed and the densification curve was gotten from the measurements of the experiments, the simulation results were compared with the results from the experiments. This comparison was done to validate the model and calibrate the

sintering parameters to sintering experiment temperatures. The calibration process used is discussed in the next chapter.

CHAPTER 4: CALIBRATION PROCEDURE AND METHODOLOGY SUMMARY

The calibration procedure was done using a MATLAB algorithm. Later this algorithm was combined with the data analysis script in python for efficiency. A kriging model was used to generate an array of possible constants to aid in the calibration process. Once good constants were found, the simulation was calibrated to experimental results. All of these steps are discussed in this chapter. The chapter concludes with an overall summary of the project methodology.

4.1 Calibration process

The constants used in the PFM simulation are temperature and time dependent and have simulation units. To map these constants to physical units a time calibration must be done, matching the simulation time in units of time steps to experimental time in units of minutes. The simulation was calibrated against experimental data by plotting the percentage change curve from the experiments against that derived from the simulations. The first step for this calibration was arbitrarily changing the values of the simulation constants related to diffusion and energy to get a good comparison between the density curves from the simulations and that of the experiments. These constants were decided with the help of a kriging model discussed next.

4.1.1 KRIGING MODEL

Kriging is an interpolation model used to approximate the response data from a deterministic computer simulation [32]. A kriging model was used to determine a viable pool of constants to use for the calibration process. The kriging model was built to imitate the sintering simulation. This was necessary because the sintering simulation takes a long

time to run. The sintering simulation takes about thirty minutes to run to about 3.5 million time steps. There are nine input constants to the sintering simulation (*con.gbmobility*, *con.surfdif*, *con.gbdif*, *con.voldif*, *con.surfenergy*, *con.gbenergy*, *con.A*, *con.B* and *con.Cgbe* in Table 2.6). *Con.gbmobility* and *con.Cgbe* were fixed to reduce the inputs to the kriging model and three of the input constants are set as dependent (diffusion coefficients) which reduces to five input constants. To further improve the results of the kriging model, *con.gbenergy* and *con.B* were set to varying between two values so that only three inputs (*con.surfdif, con.A and con.surfenergy)* are needed for two specific cases.

Next the outputs from the sintering simulation for the first 3.5 million time steps were fit to a 6th degree polynomial having seven coefficients. A seventh-degree polynomial is the highest possible fit with the amount data from the sintering simulation, but a 6th degree was chosen because very little new information was gained with a higher degree. The polynomial coefficients (outputs) and the simulation constants (inputs) were used to train the kriging model. So, the trained kriging model takes three inputs and spits out seven constants, for each predicted set of input test cases. The seven constants output from the kriging model are the predicted coefficients of the 6th degree polynomial. The polynomial fit from the kriging model results were plot against the polynomial fit from the simulation data and compared to the sets of desired constants to determine constants to put into the actual sintering simulation. For example, if the desire is to generate constants that would fit a 500 ˚C experimental run and all constants tried in the sintering simulation give rates of densification significantly lower than the experimental run, the cut-off set of constants would be the constants which correspond to the highest rate of densification and the goal of the kriging model would be to predict constants which would give rates of densification higher than the cut-off set of constants. If on the other hand, the 500 ˚C

experimental run densification curve is bounded between the densification curve of constants A and B, then the cut-off set of constants would be constant sets A and B, and the goal of the kriging model would be to predict constants which would generate a rate of densification between the results of A and B. An example of the use of cut-off constants in this kriging model is shown in the graphs in Figure 4.1. The bold dash lines in the plots correspond to the cut-off constants and the thin solid lines correspond to the predicted values from the kriging model which meet the desired criteria. In Figure 4.1a, the criterion is that the densification curves fit between the constants (bold dashed lines) and Figure 4.1b, the criterion is that the proposed densification curves have to be higher than the cut-off constants.

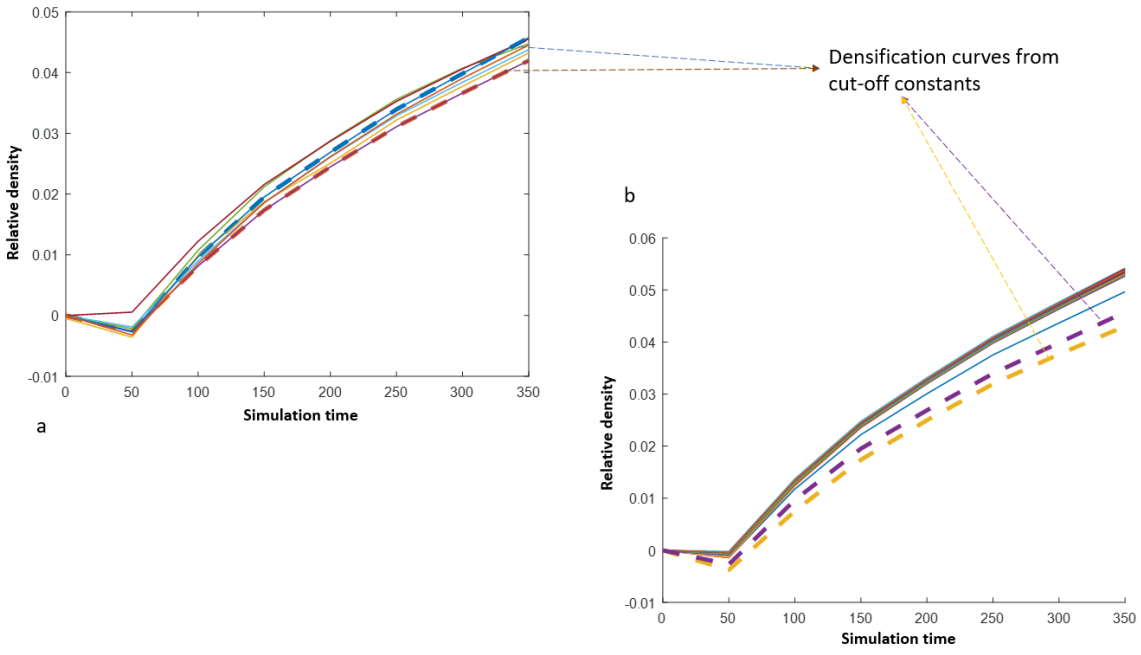


Figure 4.1: Illustration of cut-off constants and kriging goals

There are a number of sources of error with this method. Some of which include the reduction in the set of input constants used in the kriging model and the amount of sintering simulation data used to train the model. To improve on the results of the kriging model, the model was trained with an increasing number of points, using predicted points from the kriging model as inputs to the sintering simulation and then feeding the simulation results back in as additional points to use in training the kriging model. Also, the kriging model was only tested to 3.5 million time steps, which meant that there was really no good way of predicting what would happen after this milestone. Still, even with these shortcomings, the kriging model allowed for testing the viability of millions of constants which, due to time constraints, would have been impossible to do with the actual sintering simulation. Promising constant combinations were discovered using this method.

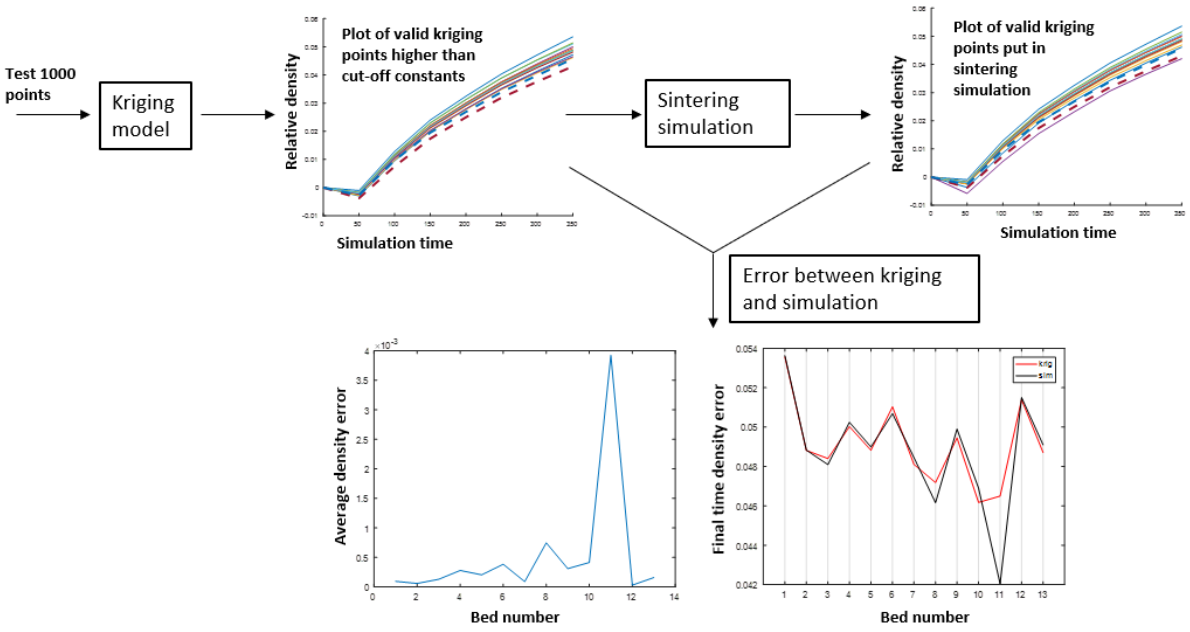


Figure 4.2: Error comparison between kriging model and sintering simulation

The Figure 4.2 shows example results of the kriging model compared to the results of the sintering simulation. The average error plot is for each of the thirteen constant sets considered. This is the average over the simulation time range from 0 to 350. The end point rho comparison in the second bottom plot compares the density value at simulation time 350 for all thirteen points. There is no in-depth analysis done in the results section as the kriging model was an auxiliary tool used to aid in picking constants.

Some of the constants derived from the kriging process where changed using studied trends and dependencies noticed between the densification behavior and values of specific constants. The next phase of the calibration was done using sets of constants which gave a good rough fit value between the results of the simulation and experiments.

4.1.2 MATLAB ALGORITHM

After a good rough fit was derived from comparing the simulation data to the experimental data, the simulation time step was calibrated to match experimental sintering time. For the time calibration, the simulation time steps are taken to have a linear correlation with the actual time so that the calibration factor is a constant (A) given in Eq. 30.

$$t_{exp}(min) = \frac{t_{sim}(timesteps)}{A} \tag{30}$$

The calibration factor was determined through a bracketing minimization algorithm in MATLAB. This algorithm works by narrowing the possible values of A to a window with upper and lower bounds off by a negligible value of epsilon. The values in this window minimize the total error between the simulation data and the experimental data. The value

of A that gives the lowest error is taken as the calibration factor mapping simulation time steps to experimental time.

4.2 Overall project methodology flow

The overall modeling approach used for this simulation is summarized in Figure 4.3.

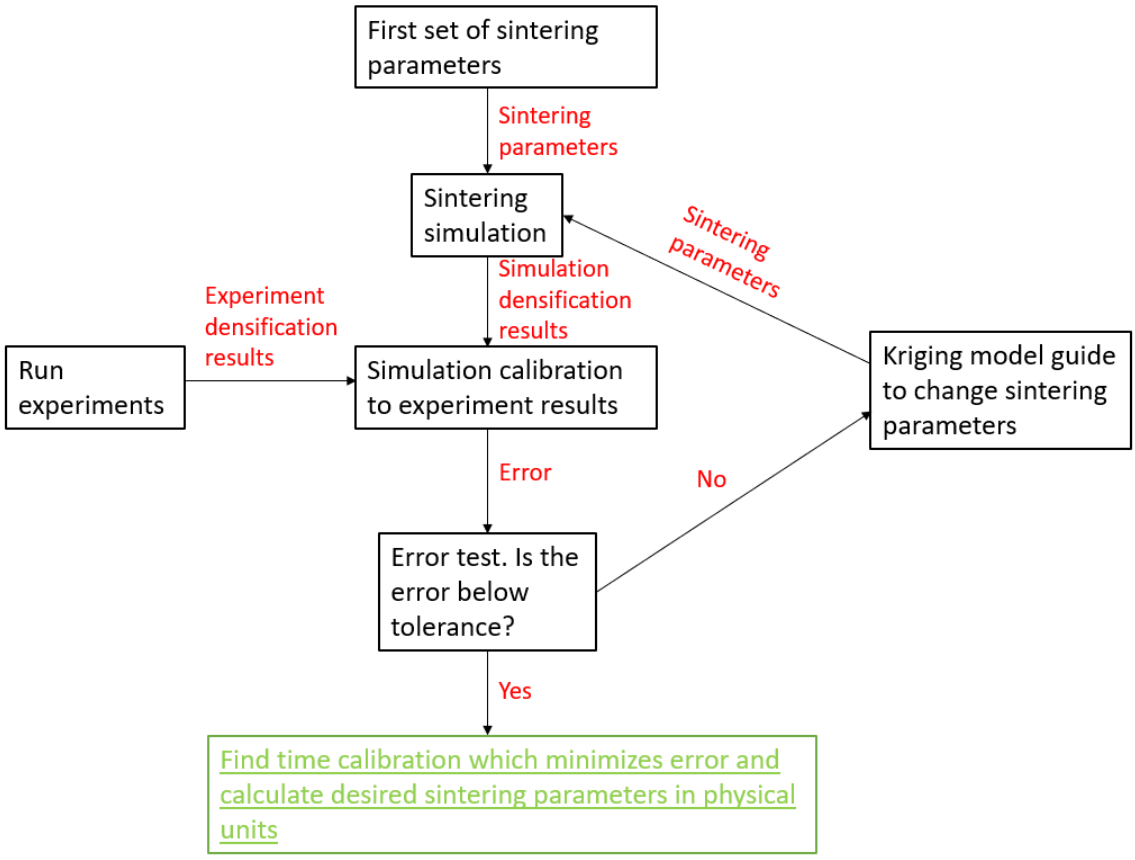


Figure 4.3: Flow chart summary of overall project methodology

The red words in Figure 4.3 correspond to the outputs going out of the black box where the arrow emanates from and inputs into the black box where the arrow terminates. The green

box at the end marks the termination of the methodology and the black boxes represent the different parts of the methodology section discussed up to this point in the paper (Chapters 2 to 4). The first set of sintering parameters are chosen arbitrarily.

The methodology summarized here is used in the project and the results are discussed in the following chapter.

CHAPTER 5: RESULTS[3]

This chapter presents the results of the project. The methodology outlined in chapters 2 to 4 was followed for the project and the results are presented here. First the results for the experiments are reported, then the simulation results as well as the results from calibrating the simulation to the experiment are discussed. Also, the mismatch in the sizes of the simulation boxes and the experiment crucibles are explained.

5.1 Experiments

The major unforeseen problem with running the experiments was the sensitivity to sample preparation. During the initial runs of the simulation, there was a lot of discrepancy in the final morphology of the samples after the experiments, which were due solely to the state the samples were in prior to sintering. Depending on the degree of drying of the ink, the results varied drastically, from fully sintered solid blocks after sintering to blocks which fell apart once removed from the crucible all the way to amorphous blobs. Examples are shown in Figure 5.1. With the amount of data needed, it was impossible to prepare every sample at once to run on the same day, so the sample preparation proved very important. The sample preparation procedure was fixed to the steps stated in the experimental procedure chapter (Chapter 3) for the results presented in this chapter.

[3] Dibua OG, Yuksel A, Roy NK, Foong CS, Cullinan M. Nanoparticle Sintering Model: Simulation and Calibration Against Experimental Data. ASME. J. Micro Nano-Manuf. 2018;6(4):041004-041004-9. doi:10.1115/1.4041668.

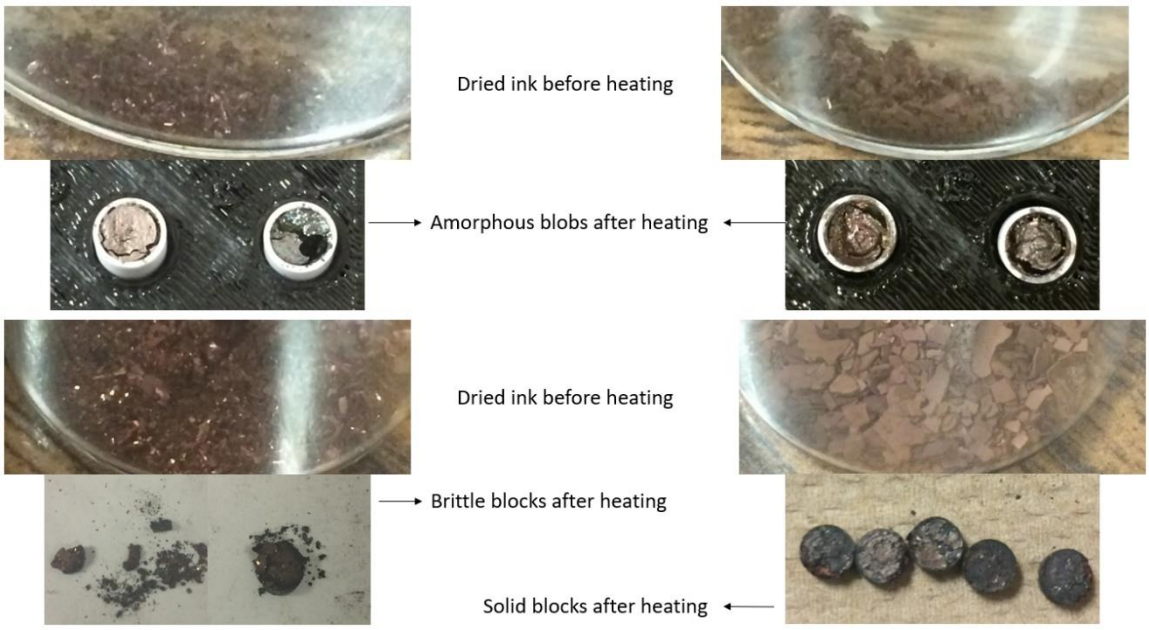


Figure 5.1: Differences in final results due to changes in drying process

5.1.1 TGA HEATING

The experiments were initially conducted in a TGA. The working procedure of the TGA introduced benefits such as lower human error and more time efficient experiment runs as a result of the automation of the process as described in the methodology section 3.2.1. However, the biggest disadvantage with using the TGA was the difficulty in accounting for cooling down time and the added heating time. Due to the automated process, the sample was kept in the TGA while the oven gradually heated up to the desired temperature and also while the oven cooled down to a temperature low enough to remove the sample, the associated cool down time was unknown and most likely variable. This variability in cool down time added an extra uncertainty in the repeatability of data obtained. Also, the heating as well as the cool down time introduced an added deviation from the simulation runs where the beds are modeled as being instantaneously, rather than

gradually, introduced to the higher temperature. An additional challenge with using the TGA was the amount of oxidation noticed in the samples tested. The coloring of the pellets in Figure 5.2 is an indication of the oxidation in the system. The images in Figure 5.2 show the samples before and after sintering.

Figure 5.2: Before and after sintering with the TGA

Analysis carried out on these experiments led to the compilation of results shown in Figure 5.3.

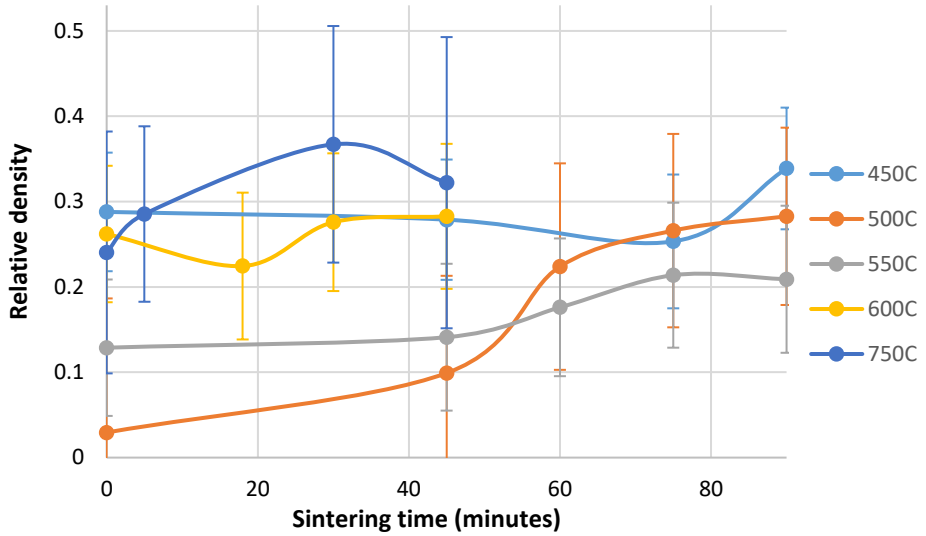


Figure 5.3: Measurement results of the TGA experiments

The data in Figure 5.3 can be broken up into two categories, temperatures at 600 ˚C and above and temperatures below. At 600 ˚C and above the experiments stop at 45 minutes because the data looks to have reached steady state at a heating time of 0 minutes. A heating time of 0 minutes corresponds to 0 isothermal heating in Figure 3.2, this however includes the heating time up from room temperature to the desired temperature. At heating temperatures below 600 ˚C the data does not make sense. The relative density at 450 ˚C for 0 minutes heating seems to have reached steady state while the relative density is a lot lower for 0 minutes heating at 500 ˚C. We would expect higher temperatures to sinter at a faster rate, as such these results don't match up with expected trends. Errors in this data most likely come from the additional heating times as discussed above, the unquantifiable degrees of oxidation during experiments and from variabilities in the sample preparation explained earlier. Experiments run with the TGA proved too inconclusive for comparison against the simulation, so the experiments were moved to a furnace.

5.1.2 FURNACE HEATING

The major advantage with using the furnace over the TGA was the control that could be achieved with respect to the heating and cooling times. The furnace allowed for the temperature to be set with the sample outside the oven while the oven heated up to the desired temperature. Then the sample was inserted once the oven was at the desired temperature, heated for the desired amount of time, and then taken out. With the sample out, the oven was cooled down. This removed the unknown heating and cooling cycles from the experiments and ensured that the experiments more closely matched the simulations. However, the drawback to this process was the amount of human error introduced in loading times and location of the crucible in the oven, as the samples were loaded manually. There was also the additional cost of timing as each experiment for a different time and temperature had to be set independently. So, before each experiment, the furnace had to be purged and then refilled with the desired gases. An additional benefit with using the furnace, revealed after the first run, was that there was a significant decrease in the amount of oxidation. This reduction in oxidation is shown in Figure 5.4.

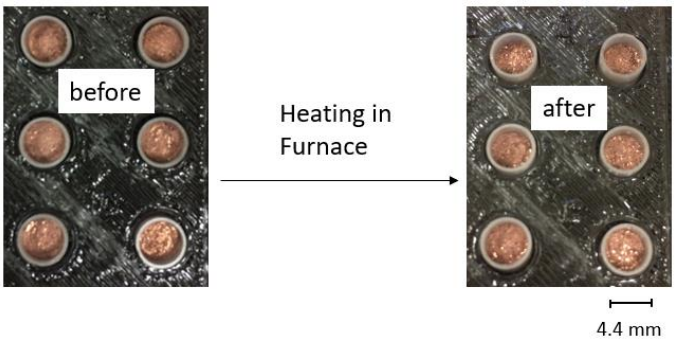


Figure 5.4: Before and after sintering with the Furnace

The images in Figure 5.4, show the crucibles before and after sintering. The pellet in the image after sintering is orange in color, which is a lot closer to the color of pure bulk copper than the TGA results in Figure 5.2.

Scanning Electron Microscope (SEM) images were taken of the flakes before sintering, and the pellets after sintering. These images in Figure 5.5, show that before sintering the particles are discrete and can be seen separate from each other. After sintering, the images of the pellets show that necks have formed between the nanoparticles after heating.

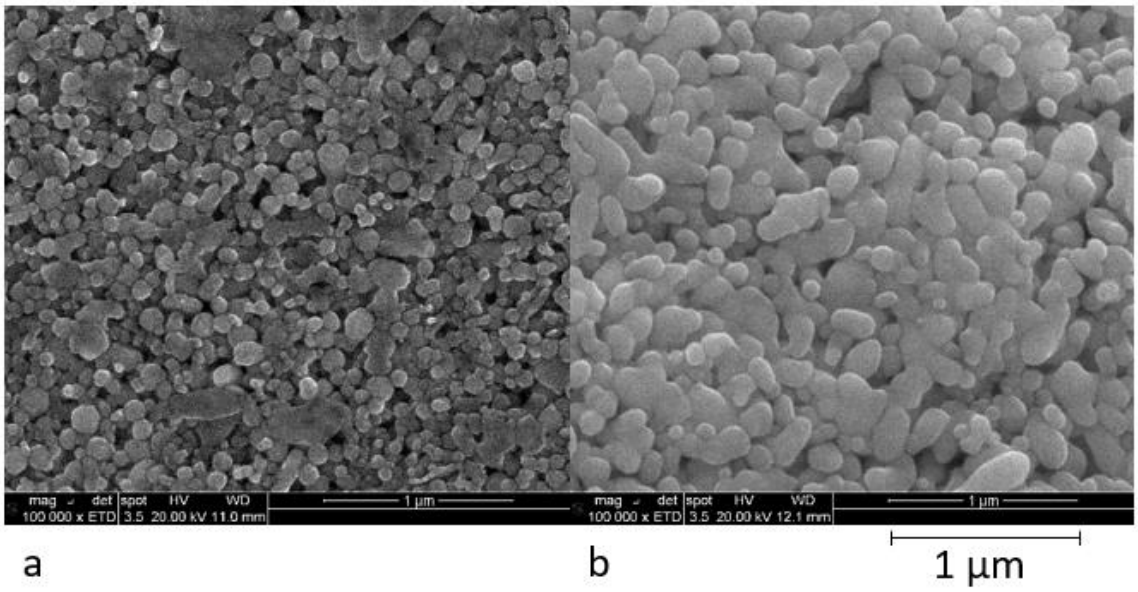


Figure 5.5: SEM Images of sintered nanoparticles. a) Before sintering. b) After sintering

The furnace sintering experiments were carried out at 450, 500, 550 and 600˚C. The data derived from these experiments are shown in Table 5.1.

450C		500C	
time (minutes)	relative density	time (minutes)	relative density
1	0.14 ± 0.05	1	0.17 ± 0.05
2	0.22 ± 0.07	2	0.24 ± 0.05
4	0.27 ± 0.05	4	0.28 ± 0.05
5	0.28 ± 0.07	5	0.28 ± 0.06
7.5	0.32 ± 0.05	7.5	0.33 ± 0.04
10	0.36 ± 0.05	10	0.35 ± 0.05
30	0.36 ± 0.05	30	0.38 ± 0.04
45	0.38 ± 0.04	45	0.38 ± 0.05
550C		600C	
time (minutes)	relative density	time (minutes)	relative density
1	0.21 ± 0.05	1	0.23 ± 0.04
2	0.28 ± 0.05	2	0.29 ± 0.04
4	0.28 ± 0.05	5	0.33 ± 0.05
5	0.3 ± 0.05	7.5	0.34 ± 0.05
7.5	0.35 ± 0.05	15	0.36 ± 0.04
10	0.34 ± 0.04	45	0.37 ± 0.04
30	0.37 ± 0.05		
45	0.38 ± 0.04		

Table 5.1: Furnace heating experiment data

The relative density data at these temperatures were fit to an exponential decay of the form in Eq. 31.

$$\rho_{rel} = K_1 e^{-\frac{K_2}{t+K_3}+K_4} + K_5 \tag{31}$$

Where the K values are best fit constants. The plot of the experiment data points and the best fit curve for the decay in Eq. 31. is shown in Figure 5.6. A consolidation of all these plots is shown in Figure 5.7. From Figure 5.6, it can be seen that sintering is characterized by an initial rate of rapid densification and as the sintering time proceeds the rate of

densification approaches a steady state value. Figure 5.7 shows that as the temperature increases, the amount of time it takes for the relative density to reach steady state decreases.

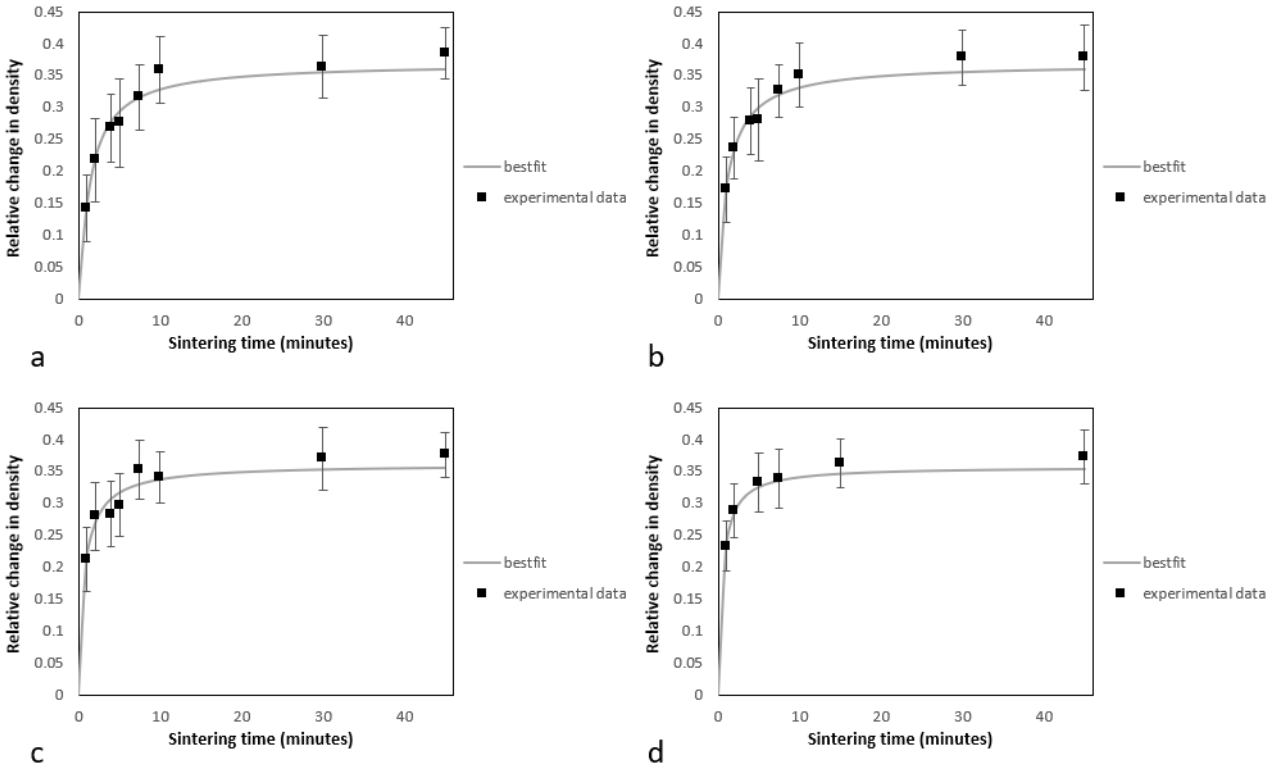


Figure 5.6: Experimental data and curve fit at a) 450˚C. b) 500˚C. c) 550˚C. d) 600˚C

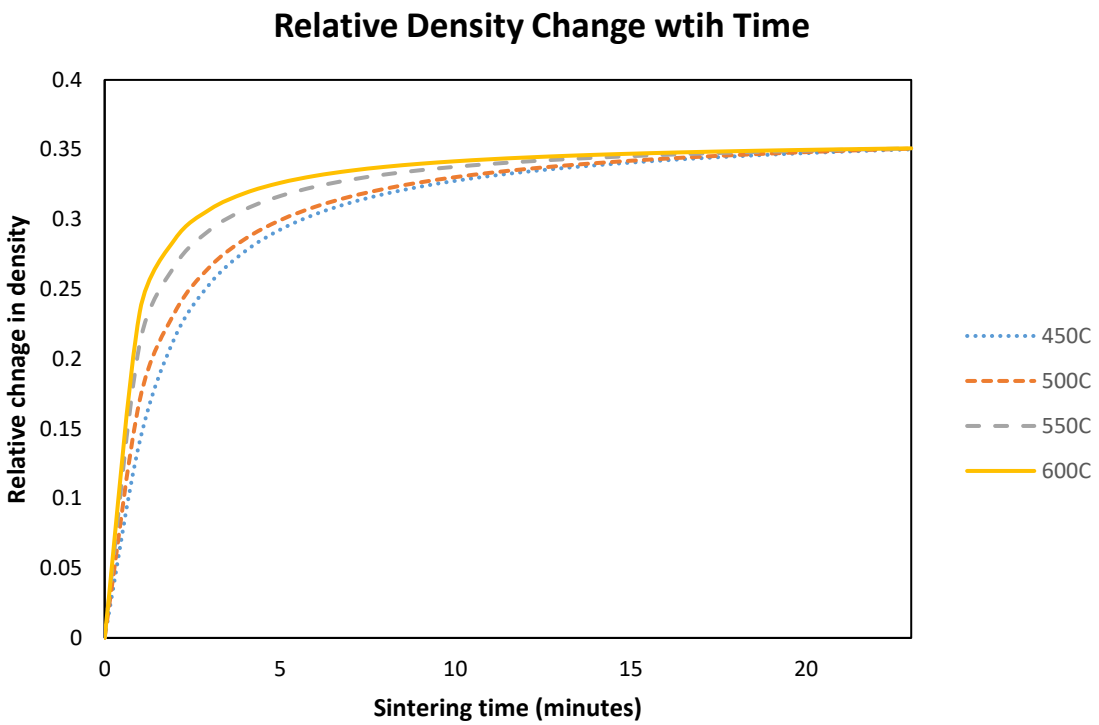


Figure 5.7: Consolidation of experiment fit plots

Comparing the densification curves in Figure 5.3 from the TGA and Figure 5.7 from the furnace heating, it is clear that the data from the furnace heating is more accurate as it conforms to expected trends in Figures 5.6 and 5.7. As such, the curves from the furnace experiments were used in the simulation calibration. The results presented in this section were set against the results from the simulation and the appropriate sintering parameters and simulation time calibration constant were calculated. The process of deriving this is discussed in section 4.1. The results from the simulations are presented next.

5.2 Simulations

The results of the simulation are presented here. First the results for the 2D simulations are presented. Then the simulation is transitioned from 2D to 3D and the results of the 3D simulations are presented.

5.2.1 2D RESULTS

The simulation was initially developed and tested with a 2D system. The 2D system was used to visually test the accuracy of the densification in the C++ code. The first test was done with a 2 particle system and then a 3 particle system. The 2 and 3 particle systems required the specific definition of circle information for each particle at the initiation of the box, and the particle specific eta variables during the iteration. This required writing three lines of code for each particle definition (radius, x and y center) and about five lines of code for the iteration (defining the sub-variables needed for the next eta value eta($t+\Delta t$)). These manual entries were infeasible for a large number of particles. As such, the simulation was transferred from a few particles to many particles, in preparation for the 3D system. The manual definition related to particle specific terms is taken out. Particle initiation is done by reading into the simulation from a file containing the particle information which is tested for different arrangement and sizes of multiple particle beds. Also, the evolution of the eta variables is done using a number of for loops to automatically calculate each new eta value. The results from this transition are presented in this section.

5.2.1.1 2 particles

The first test of the simulation was done with a system with 2 particles. The results of these simulations are shown in Figure 5.8. The simulation uses two equal size particles

both having a radius of 20 pixels in an 85 by 42 simulation box. The system is initiated with the particles just touching that is, there is no overlap between them.

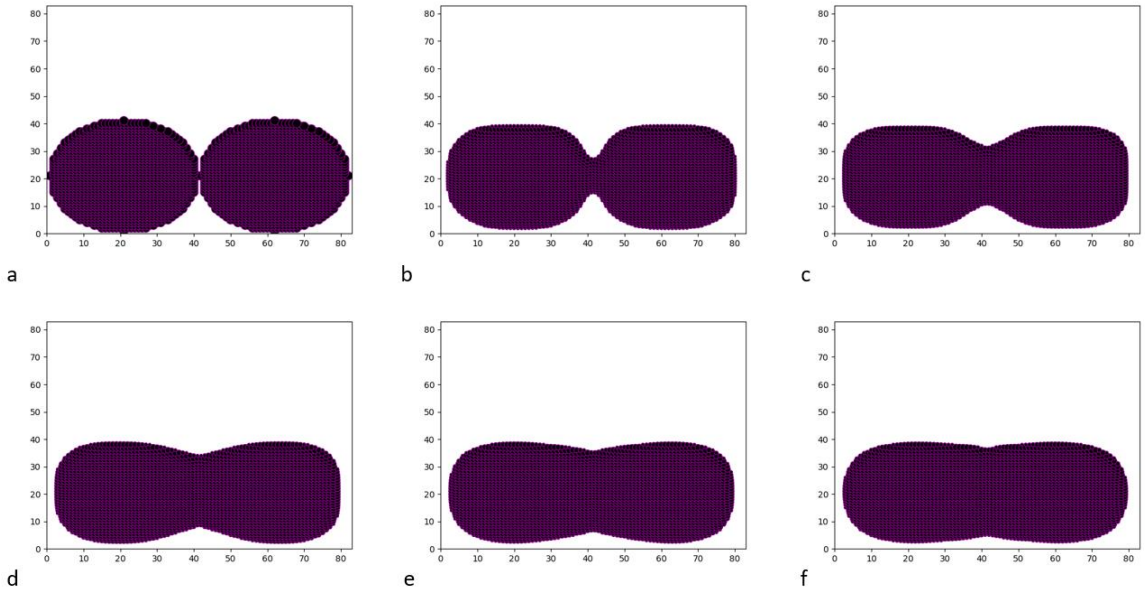


Figure 5.8: The evolution of a 2 particle system in 2D a) Initial b) 1000 timesteps c) 10000 timesteps d) 25000 timesteps e) 50000 timesteps f) 100000 timesteps

In Figure 5.8, as the sintering starts, a neck forms between the particles. The pixels diffuse across the grain boundary and along the surface filling the neck. There is clear shrinkage along the x and y bounds as sintering occurs.

5.2.1.2 3 particles

After the 2 particle test, a test was conducted with a 3 particle system. Figure 5.9 shows the evolution of this 3 particle system.

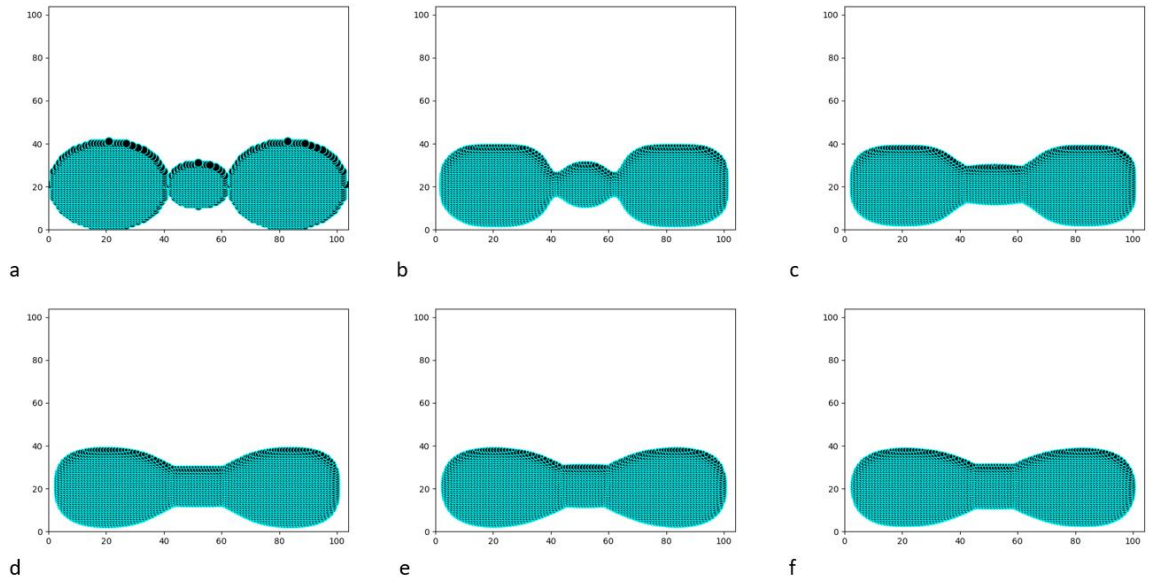


Figure 5.9: The evolution of a 3 particle system in 2D: 20-10-20 pixel diameters. a) Initial b) 1000 timesteps c) 10000 timesteps d) 25000 timesteps e) 50000 timesteps f) 100000 timesteps

Figure 5.9 shows the evolution of a 3 particle system in 2D. The system uses two 20 pixel radii particles and a 10 pixel radius particle in between both of them. The effect of changing the sizes of the particles on the system is tested and the results are shown in Figure 5.10.

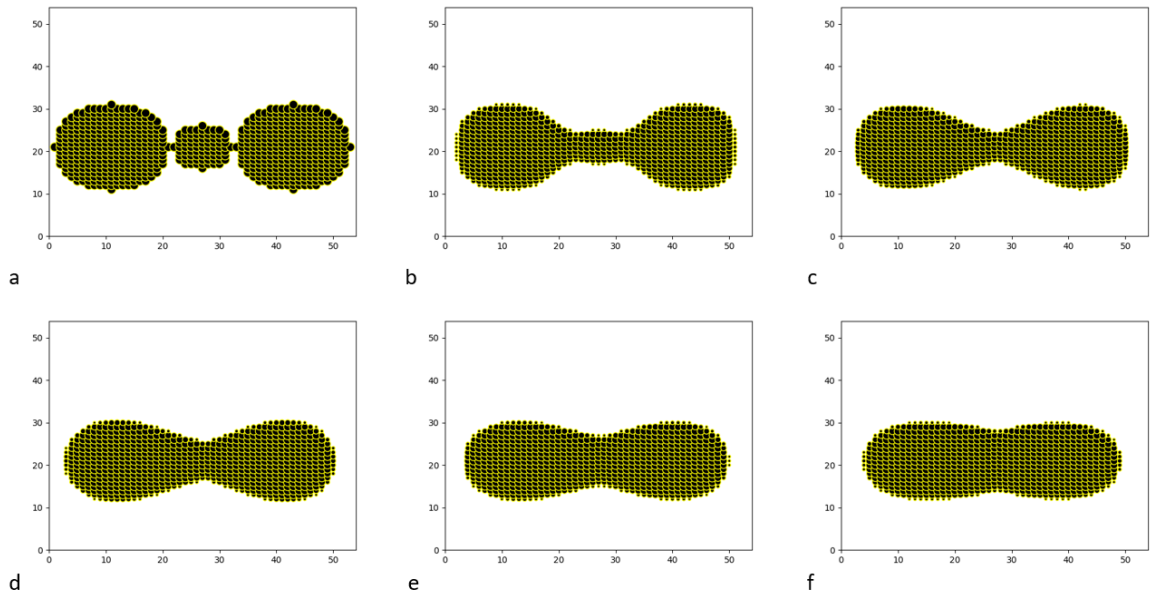


Figure 5.10: The evolution of a 3 particle system for smaller particles: 10-5-10 pixel diameters. a) Initial b) 1000 timesteps c) 10000 timesteps d) 25000 timesteps e) 50000 timesteps f) 100000 timesteps

The system in Figure 5.10 consists of two 10 pixel radii particles and a 5 pixel radius particle. In the system, the grain boundaries of the larger particles move into the smaller particle at a faster rate than the system in Figure 5.9. These configurations were done with the default constants in Table 2.6.

Additionally, the effects of changing the sintering parameters in Table 2.6 is investigated. The rate of surface to grain boundary diffusion constants is studied first. When the grain boundary diffusion is decreased so that the ratio of surface to grain boundary diffusion is 100:1, the rate of sintering decreases. The microstructural evolution looks the same, but the amount of time it takes to reach these stages is considerably longer. These results are shown in Figure 5.11.

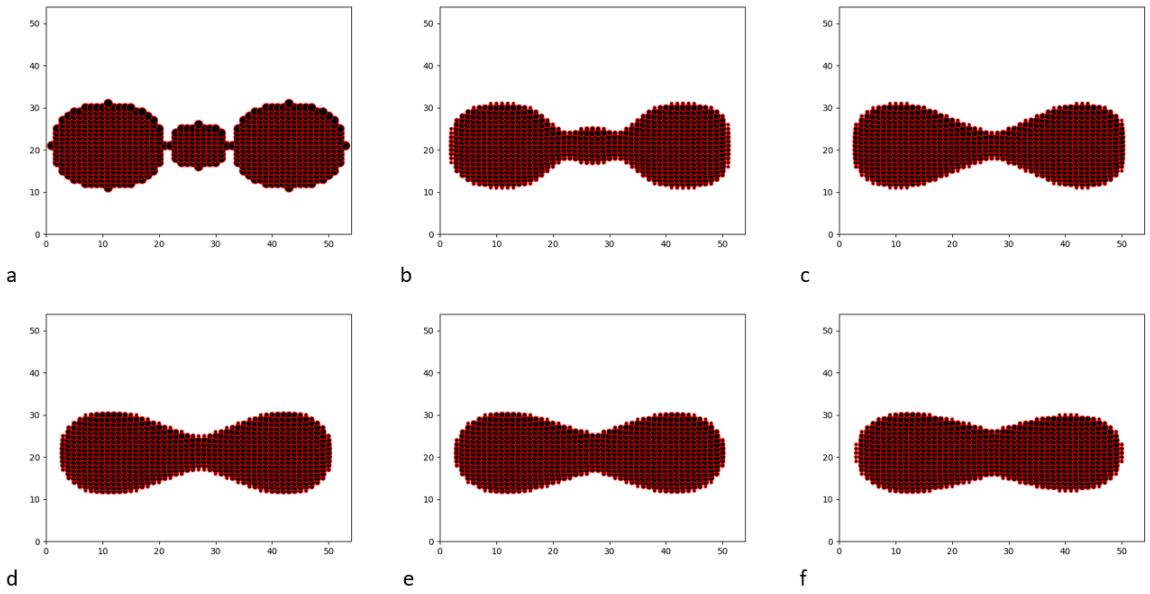


Figure 5.11: Effects of reduced grain boundary diffusion. a) Initial b) 1000 timesteps c) 10000 timesteps d) 25000 timesteps e) 50000 timesteps f) 100000 timesteps

When the grain boundary is increased so that the ratio of surface to grain boundary diffusion is 50:1, the rate of sintering speeds up.

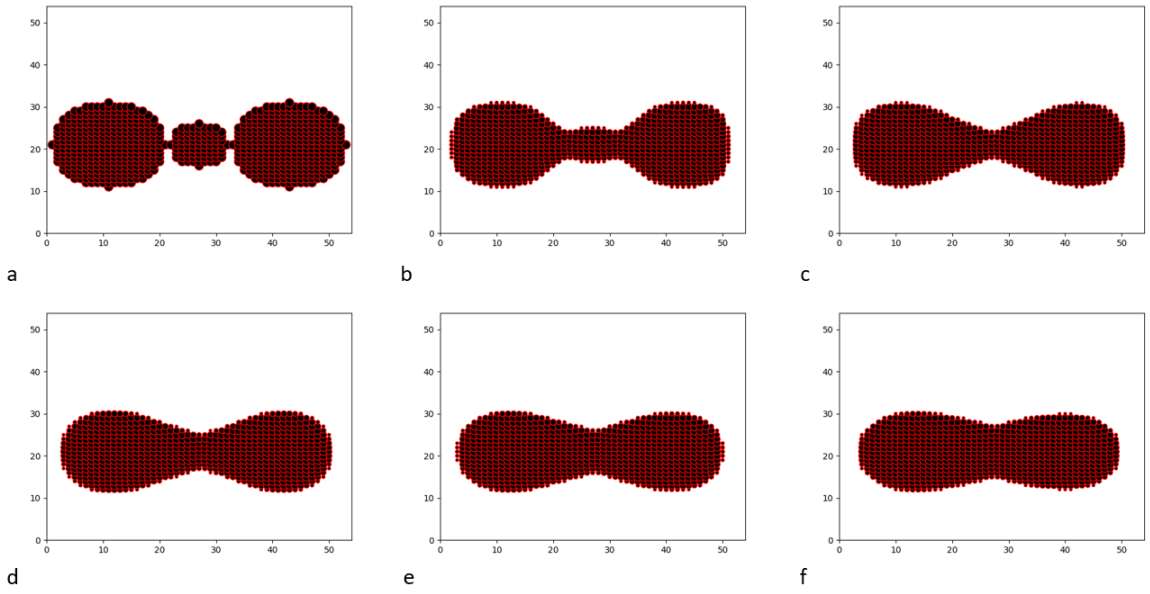


Figure 5.12: Effects of increased grain boundary diffusion. a) Initial b) 1000 timesteps c) 10000 timesteps d) 25000 timesteps e) 50000 timesteps f) 100000 timesteps

These results show that the diffusion constants have a significant effect on the rate of densification. While the morphology of the system evolution remains very similar, with the same stages of rapid grain boundary migration and filling of the neck between the two particles formed, the rate changes.

After changing the diffusion coefficients, changes in the ratio of surface to grain boundary energy is studied. The values of the sintering parameters *con.gbenergy* and *con.surfenergy* are changed. These parameters are related to surface and grain boundary energy as in Eqs. 3 and 4. The results of changing these constants are shown in Figure 5.13.

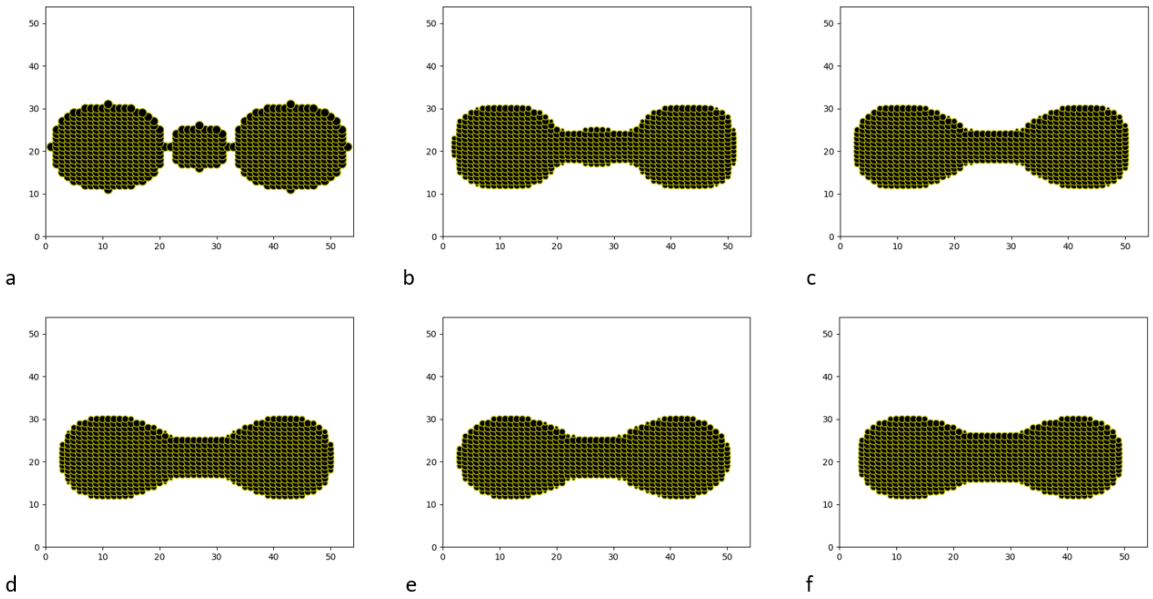


Figure 5.13: Changing the ratio of surface to grain boundary energy. a) Initial b) 1000 timesteps c) 10000 timesteps d) 25000 timesteps e) 50000 timesteps f) 100000 timesteps

The results in Figure 5.13 correspond to a reduction of the grain boundary energy as well as an increase in the surface energy in the system. The previous results correspond to systems with surface to grain boundary energy ratios of 1.88:1. The ratio of surface to grain boundary energy in Figure 5.13 is 3.81:1. In Figure 5.13 it is clear that the rate of the grain boundary moving from the larger particles into the smaller particle is a lot slower than when the surface to grain boundary energy ratio was lower. In the case in Figure 5.13, the contribution of surface diffusion to the evolution is a lot higher than in the case in the previous figures, where in Figure 5.13, the neck between the particles is filled by the motion of pixels from the surfaces of the larger particles to that of the smaller.

These few particle simulations were a good building block to enhancing the simulation into working for higher numbers of particles. 2D results for a higher number of particles are shown in the next section.

5.2.1.3 Many particle 2D simulations

Changes in the simulation were made to make it easier to track the evolution of a higher number of particles. This was tested out first with a system of 16 equally sized particles.

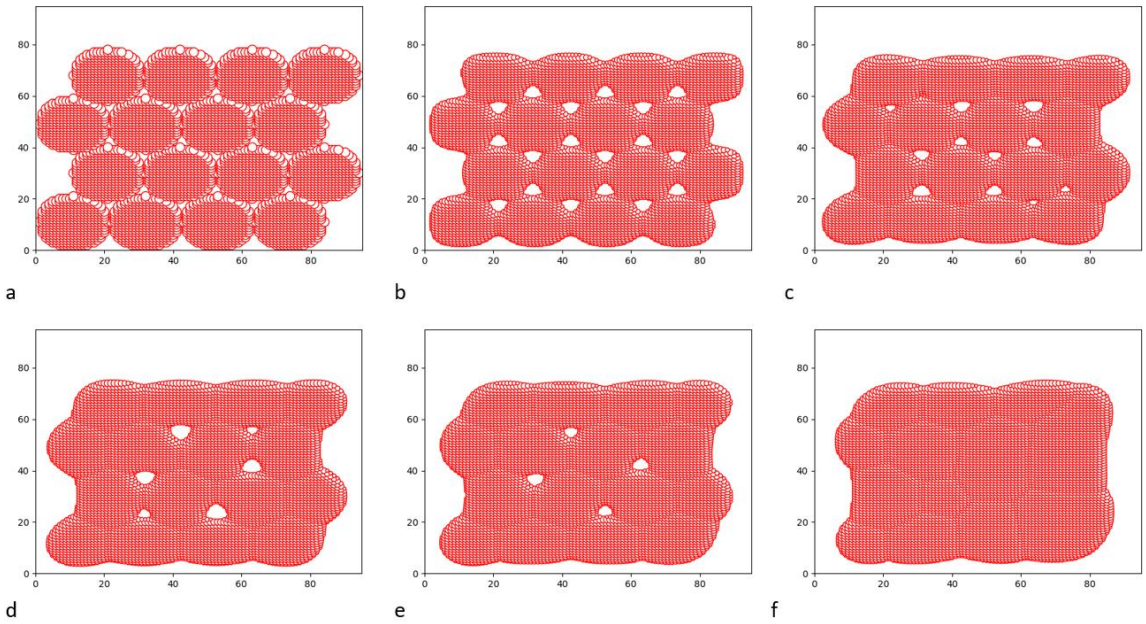
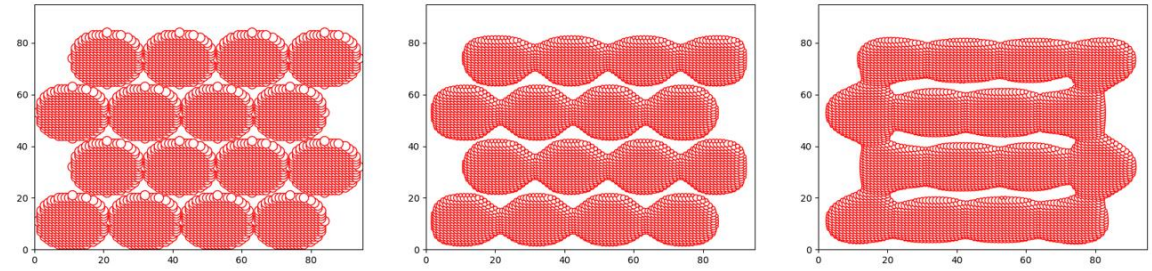


Figure 5.14: Evolution of 16 closely packed particles. a) Initial b) 1000 timesteps c) 10000 timesteps d) 25000 timesteps e) 50000 timesteps f) 100000 timesteps

Other configurations of the 16 particle systems were tried and are shown in the images below. These configurations were derived from changing the amount of overlap between rows of particles. The first configuration in Figure 5.15 has an average of 2 less pixels overlapping, and the second configuration has on average 1 less pixel overlap than Figure 5.14. The results show that with a higher value of initial overlap there is a lower degree of porosity at the final time point f (100000 timesteps). With the initial overlap in Figure 5.14, the final particle system is fully dense.


a
b
c
d
e
f

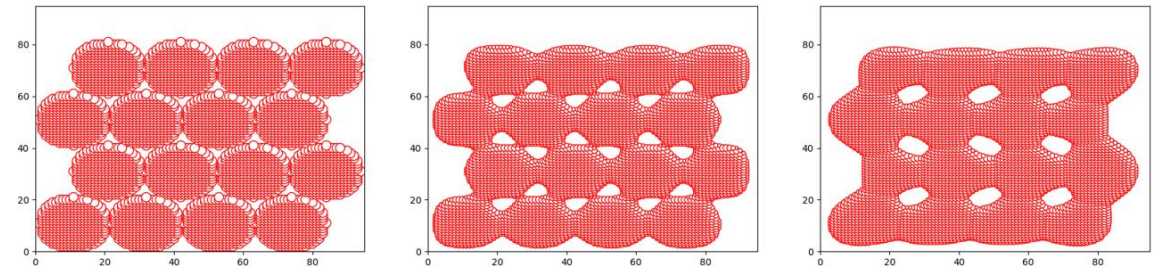

a
b
c
d
e
f

Figure 5.15: Different configurations of a 16 particle system. a) Initial b) 1000 timesteps c) 10000 timesteps d) 25000 timesteps e) 50000 timesteps f) 100000 timesteps

Each of these configurations show a clear rate of densification as time proceeds. Finally, the simulation was tested with a 59 particle system in a 150 by 150 pixel box. This box contains randomly spaced particles having radii between 6 and 15 pixels. The results are shown in Figure 5.16.

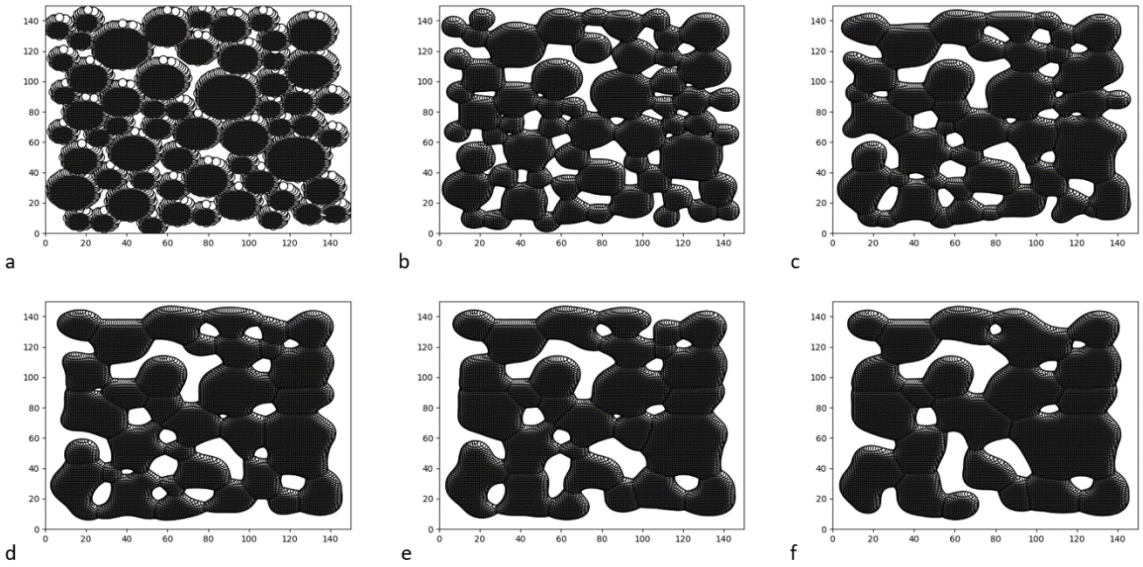


Figure 5.16: Evolution of a randomly packed system a) Initial b) 10000 timesteps c) 100000 timesteps d) 250000 timesteps e) 500000 timesteps f) 1000000 timesteps

While the 2D simulations were a great platform for creating the simulation, testing its validity, and varying the sintering parameters to gain an understanding of their effects on the sintering properties, it was impossible to use these simulations for modeling the sintering of an actual sintering bed. This is because there is no volume in the 2D cases and as such the rate of densification gotten in 2D greatly differs from the densification in 3D.

Additionally, since there is no concept of bed height, it is impossible to obtain a true picture of height shrinkage from a 2D simulation. As such, the simulation was transitioned to a 3D simulation to be able to get better predictions for the sintering behavior.

5.2.2 3D RESULTS

In this section the results for the 3D simulation are presented. First a simple case for 2 particles is presented. Then simulation beds are tested, starting with a one by one and then a two by two micrometer bed. Analysis is done on the one by one micrometer bed.

5.2.2.1 2 particles

The first step of transitioning the simulation from 2D to 3D was testing with a 2 particle system with particles of unequal size, in a 40 by 40 by 40 pixel box. For this simulation, the bigger particle was set to have a diameter of 20 pixels and the smaller particle a diameter of 10 pixels. The results are shown in Figure 5.17.

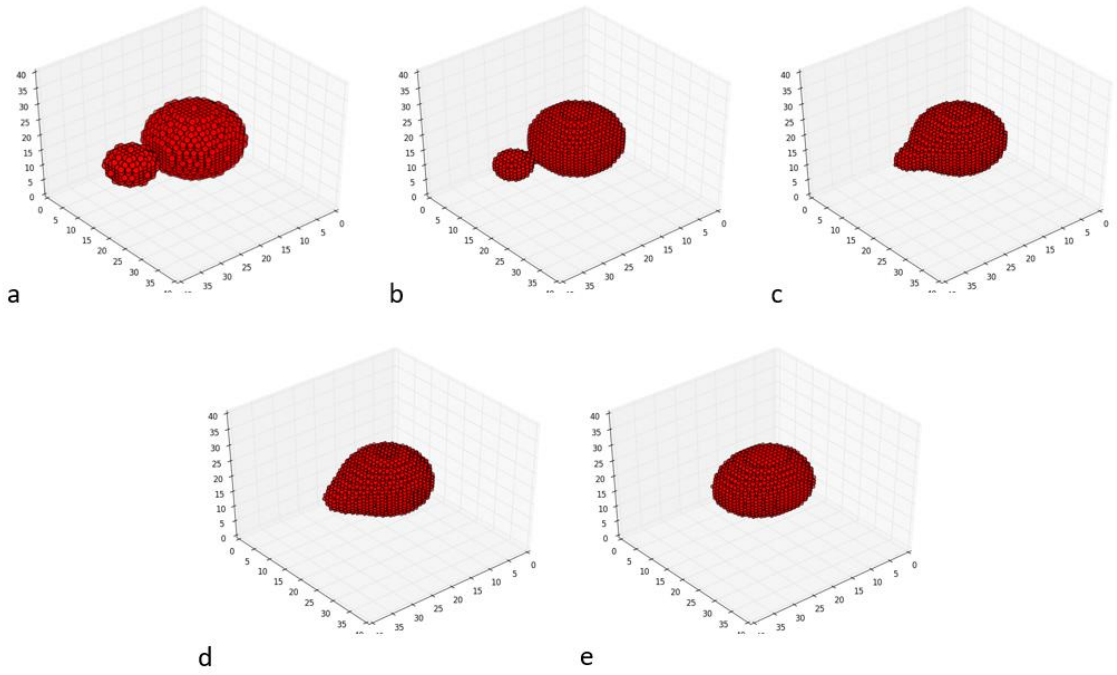


Figure 5.17: The evolution of two particles with unequal sizes during sintering. a) Initial b) 2500 timesteps c) 12500 timesteps d) 22500 timesteps e) 50000 timesteps

At the start of the process the particles are just touching to initiate diffusion. As sintering progresses a neck starts to form between particles which continues to grow until the onset of coarsening. During the process of coarsening, the boundary between the particles migrates into the smaller particle shifting the mass balance towards the bigger particle. As the simulation time increases, this shrinkage continues until only a single spherical particle remains at steady state. At 50,000 time steps the simulation is approaching steady state but has not quite reached it. The initial sintering periods are characterized by fast neck growth between particles which slows down as coarsening begins. The 2 particle simulation took 7.2 minutes to run to the final time step using a single processor. This length of time for two particles running to 50,000 time steps shows the need for parallel computing to achieve higher final times with larger size beds and higher number of particles.

5.2.2.2 One by one micrometer beds

One-by-one micrometer beds were generated with the bed generation tool described earlier and some examples are shown in Figure 5.18. The size range of these beds is designed to match the size distribution of the copper nanoparticles in the ink used for the experiments. The size distribution of this ink was derived from experiments to fit a lognormal distribution with a mean diameter of 232 nm and a standard deviation of 96 nm.

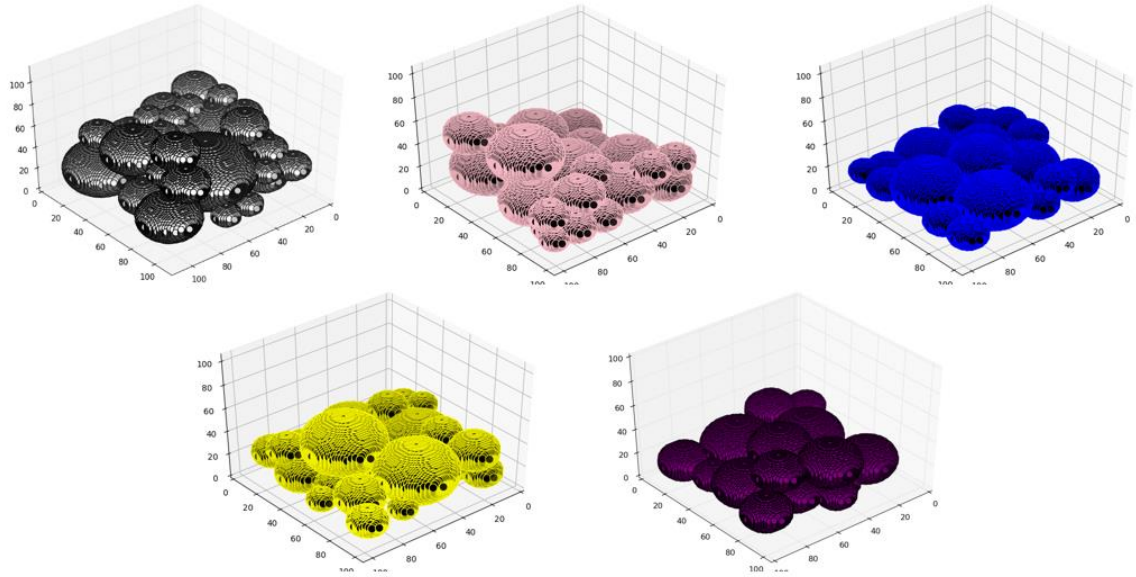


Figure 5.18: Example one by one micrometer beds generated with bed generation tool

5.2.2.2.1 Simulation results

The bed in Figure 5.19 is made up of 43 particles in a simulation box with dimensions of 110 by 110 pixels in the x-y plane and 73 pixels in the z. This corresponds to a bed height of 750 nm and particles with diameters ranging from 146 to 573 nm with a mean of 218 nm, using a conversion factor of approximately 10.6 nm/pixel. The simulation was done with this bed using the parallel version of the code. It was performed using a 64

core cluster and took approximately 19 hours to run to 2.4 million time steps on Stampede. The results of the sintering simulation are shown in Figure 5.19.

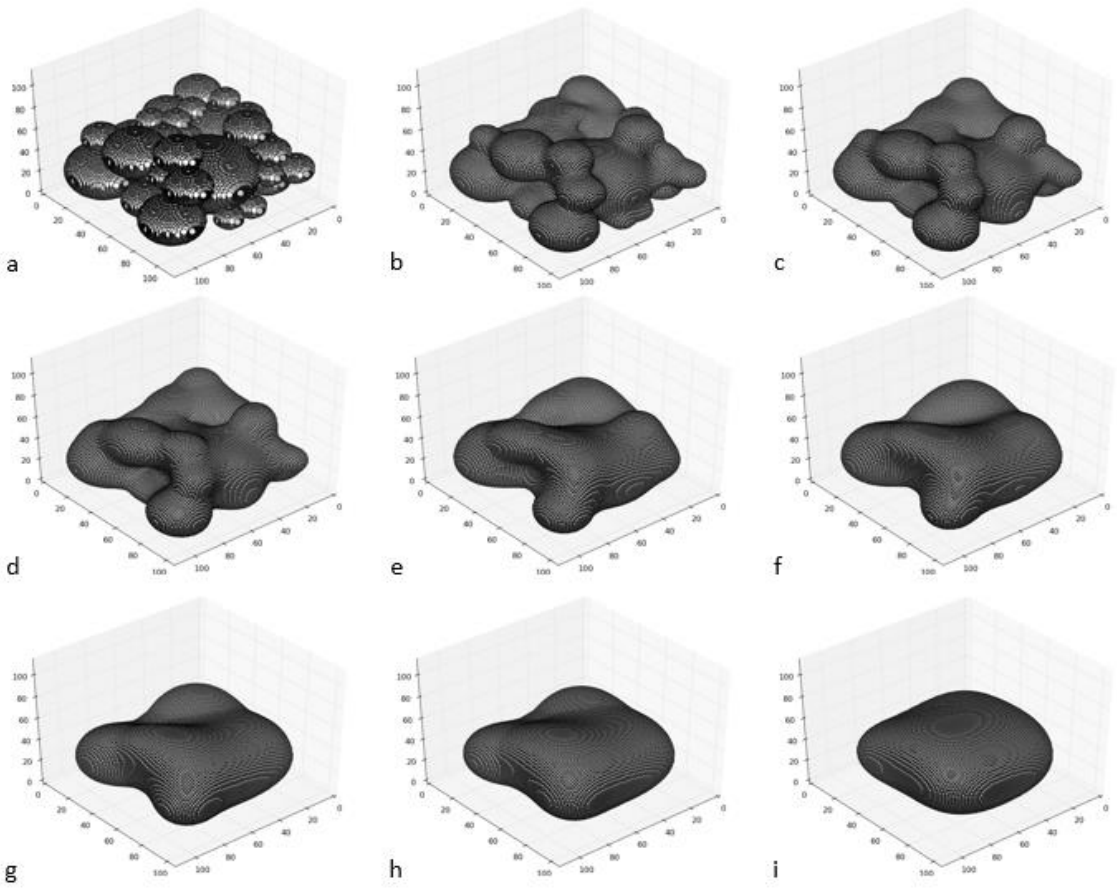


Figure 5.19: The evolution of a one by one micrometer bed. a) Initial. b) 20000 timesteps (.16 hrs.) c) 60000 timesteps (.47 hrs.) d) 100000 timesteps (.79 hrs.) e) 280000 timesteps (2.2 hrs.) f) 440000 timesteps (3.6 hrs.) g) 700000 timesteps (5.5 hrs.) h) 1100000 timesteps (8.7 hrs.) i) 2400000 timesteps (19 hrs.)

At the start of the sintering process particles are in contact to initiate diffusion. The average overlap between the particles at the start of the simulation is 0.0004 nm. The sintering behavior in this system follows the same stages as discussed for the 3D 2 particle system.

As shown in Figure 5.19 the initial sintering periods are characterized by fast neck growth evident in Figures 5.19a – d. The neck formation happens rapidly leading to a faster rate of evolution than in the later time steps. The shrinkage and densification become clearer in the later Figures 5.19f – i as the pores prevalent in the previous time steps are filled due to diffusion as sintering occurs. The bed shown in Figure 5.19 was simulated under isothermal conditions, with each particle having the same values for the diffusion and energy constants. The values in the description of Figure 5.19 are the simulation time steps and in parenthesis the amount of wall clock time it takes to get to the corresponding time step. Cross-sectional images are shown in Figure 5.20.

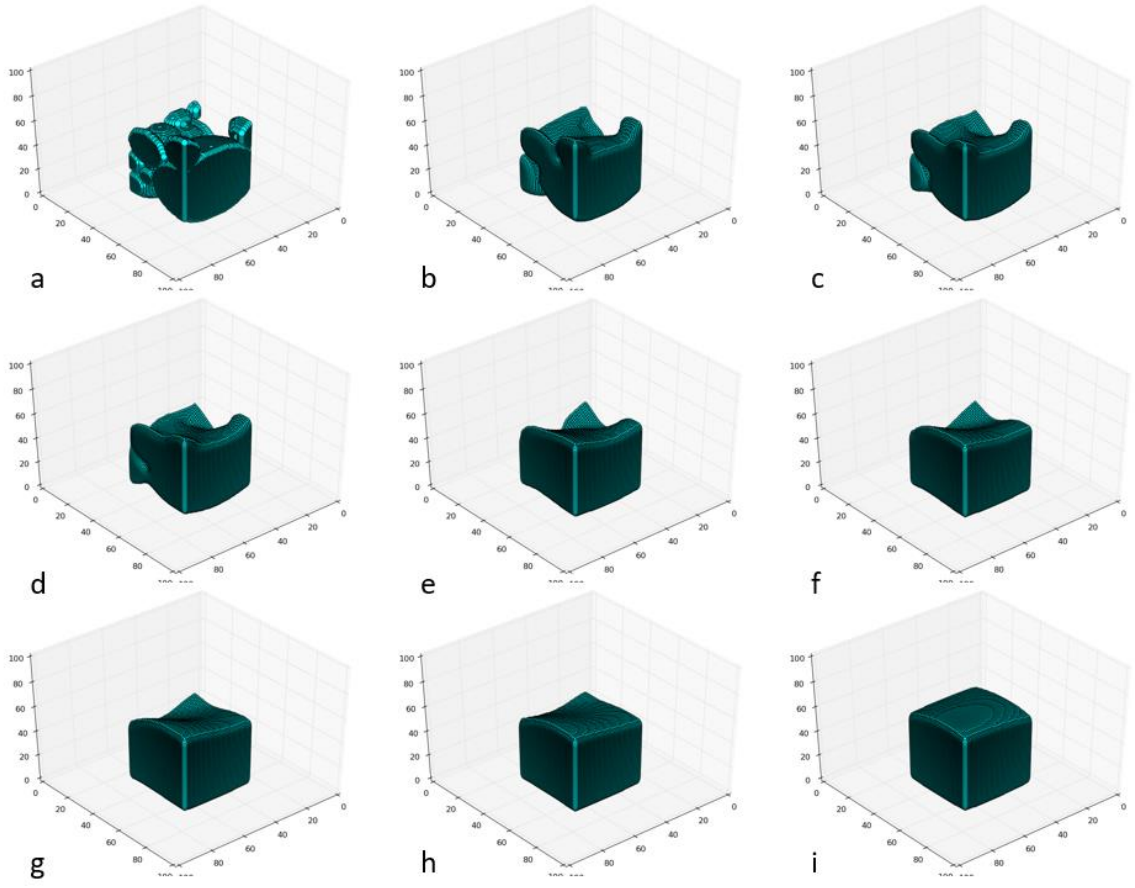


Figure 5.20: Cross-sectional images in the center of the simulation bed after a) Initial. b) 20000 timesteps c) 60000 timesteps d) 100000 timesteps e) 280000 timesteps f) 440000 timesteps g) 700000 timesteps h) 1100000 timesteps i) 2400000 timesteps

5.2.2.2.2 Data analysis

Analysis was done on the bed in the section above. The first step was determining the acceptable bounds to use for analysis. This was done using the edge finding algorithm discussed in the methodology. Images in Figure 5.21 show different analysis bounds derived for a bed based off different cutoff factors. The deviation in these beds corresponding to different cut offs is shown in the figure below.

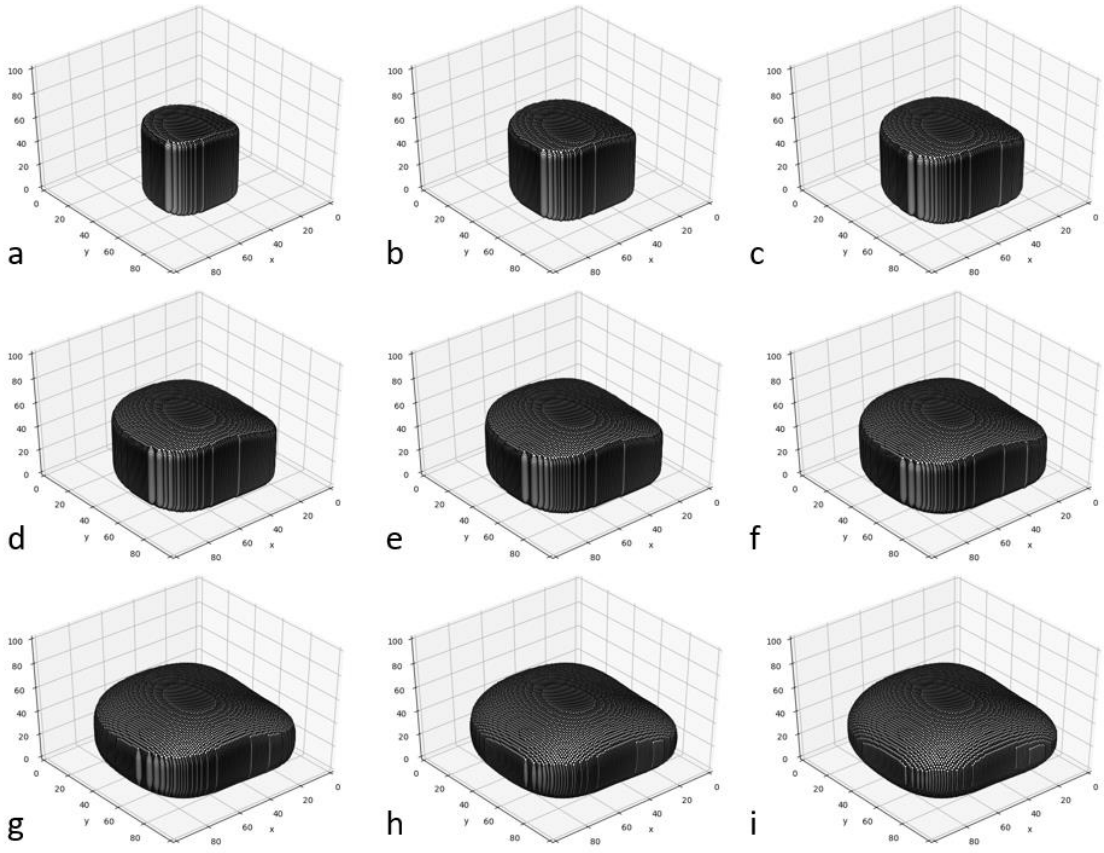


Figure 5.21: Results of edge-finding algorithm. Pixel deviation from bottom (cut-off factor): a) 0 pixels b) 1 pixels c) 2 pixels d) 4 pixels e) 6 pixels f) 8 pixels g) 12 pixels h) 16 pixels i) 20 pixels

The images in Figure 5.21 show the effect that changing the cut-off factor has on the analysis box chosen. This cut off factor is dependent on the pixel deviation from the bottom of the bed. When more deviation is allowed the analysis box gets closer to the edges of the simulation.

Analysis was done on the boxes derived from the edge finding algorithm and the results are shown in Figure 5.22. This figure shows the average rate of relative density for a single run of the simulation bed discussed above, as well as the associated error bounds calculated from varying the position of the analysis box.

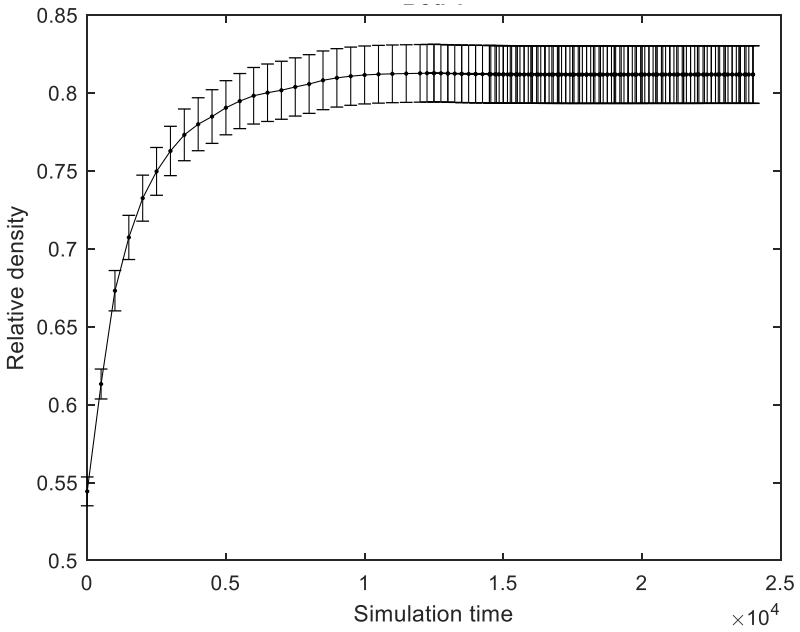


Figure 5.22: Relative density curve with error bounds

5.2.2.3 Two by two micrometer beds

One of the goals for the model described in this paper is to be able to apply it to over a hundred particles. To this end the simulation was applied to a two-by-two micrometer bed. The configuration of this bed was determined using the same bed generation tool as described earlier, and the PFM simulation was run on this bed. The two-by-two micrometer bed corresponds to 134 particles with diameters ranging between 118 to 572 nm. The bed was created using the same particle size distribution as used for the one-by-one micrometer bed. The two-by-two micrometer bed had a simulation box size of 286 by 282 pixels in the x-y plane and 98 pixels in the z. With a 140 core cluster the simulation took 48 hours to run to 220,000 time steps, and 52 hours to run to 2,030,000 time steps in a 1440 core cluster. The success with changing the number of processors used to run the simulation shows the versatility of using parallel processing with this simulation. Following this trend, the simulation can reach higher time steps with large number of processors. The results from the simulation are shown in Figure 5.23.

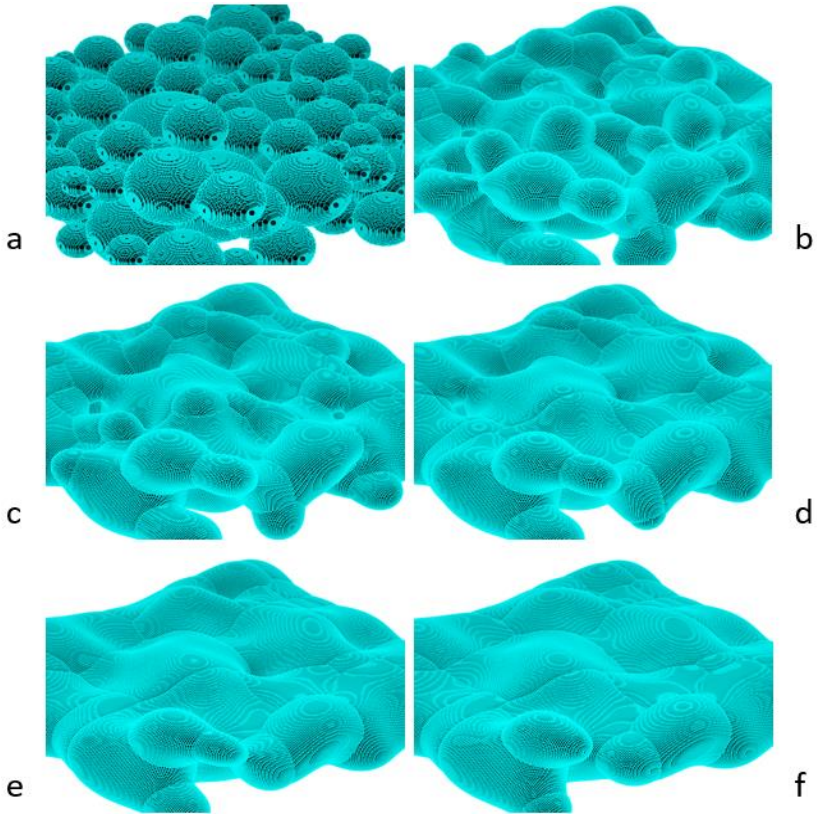


Figure 5.23: The evolution of a two by two micrometer bed with 134 particles. a) Initial b) 200000 timesteps (5.1 hrs.) c) 600000 timesteps (15 hrs.) d) 1000000 timesteps (26 hrs.) e) 1500000 timesteps (38 hrs.) f) 2000000 timesteps (51 hrs.)

As was the case with the one-by-one bed, these images show a similar rate of rapid initial neck formation and a slower rate of densification and shrinkage as sintering proceeds. Analysis done on this bed and shown in Figure 5.24 quantitatively confirms that the densification is similar to that seen in the one-by-one micrometer bed.

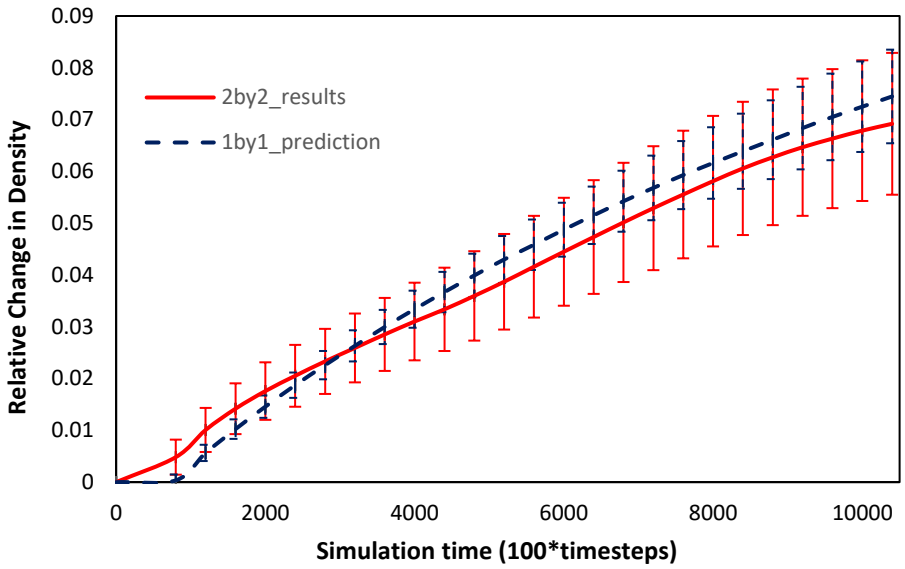


Figure 5.24: Relative change in density curve derived from data analysis done on a 2-by-2 micrometer bed and the prediction from a 1-by1 micrometer bed.

The plots in Figure 5.24 show two density curves. The density curve in the dashed line is a prediction from scaling the results of a one-by-one micrometer bed by a factor of the areal magnification between both beds. The value of the areal magnification between the beds is defined on the order of the ratio between the areas of both beds. The average error between the prediction and the actual density curve derived from the analysis of the beds is 12% which is smaller than the uncertainty in these curves. This error value shows a good degree of accuracy in scaling up the results using analysis of the smaller bed. The significance of this test is to show the viability of extrapolating results from one-by-one micrometer beds for larger beds which are more computationally expensive to run. The results obtained here suggest that edge effects are minimal due to the similarity in the results between the smaller and larger beds which have much lower surface to volume ratio than the smaller ones. However, further analysis must be done to test this hypothesis.

5.3 Simulation and experiment size mismatch

The goal of carrying out the experiments is to validate and calibrate the simulations. For this calibration, millimeter scale crucibles are used in comparison to micrometer scale simulation beds. This mismatch in size is a result of the infeasibility of applying the same size scales to both the simulation and the experiments. That is, facilities do not exist to conduct experiments on the micron scale and running the simulation on the millimeter scale would require more computational power and time than possible. As such, initial density measurements are taken to make sure that the simulation bed and the experiment bed at the start of sintering are comparable. The procedure for obtaining these density measurements in the experiments are discussed in the experimental procedure section 3.3. The initial density in the simulation bed is determined from the porosity of the simulation bed. The number of pixels in the vapor phase are counted and the pore density is obtained from taking the ratio of porous pixels to the overall number of pixels in the simulation bed. Finally, the initial density for the simulation is calculated by multiplying the percentage of filled pixels (1 – porosity) in the bed with the known density of bulk copper 0.00896 g/mm^3. Distributions for the average initial densities from the experiments and simulations are shown in Figure 5.25 with lines to indicate the 95% confidence intervals for each distribution. The measurements from the experiments are obtained from an average of 6 measurements taken for each bed. This makes the average density from the experiments follow a normal distribution with an average of 3 mg/mm^3 and a standard deviation of 0.06 mg/mm^3. The initial density for the simulations follows an unknown distribution. Based off the Central Limit Theorem the approximation can be made that the average of the initial densities from the simulation would follow a normal distribution with a mean of 3.2

mg/mm^3 and a standard deviation of 0.1 mg/mm^3 (which is the sample standard deviation divided by the square root of the number of points sampled).

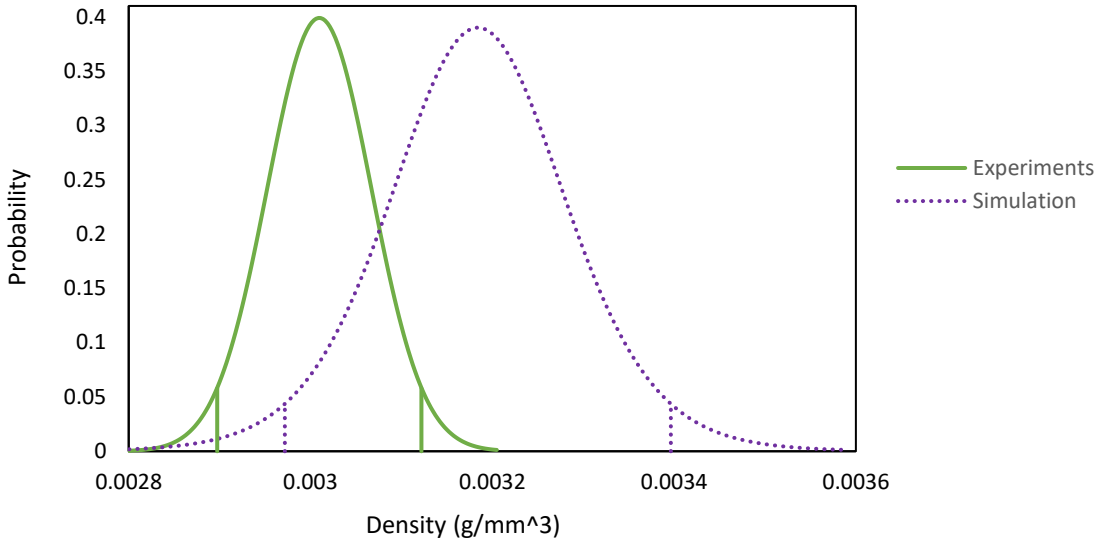


Figure 5.25: Distribution of the Average Initial Density of the Simulation and Experiment for 12 Simulation beds and 24 Experiment samples

A two-tailed p-test was done to test the null hypothesis that both distributions are equal. The p value calculated from this test is 0.074. This value is greater than 0.05, so the null hypothesis cannot be rejected at a 90% confidence level. One reason why the initial densities of the simulations and experiments may differ is due to the precision of measurement. The height measurements of the experiments are done with plastic stoppers which introduce additional errors due to the irregularities in the surface of the stopper. On the other hand, there is no such irregularity in measuring the density of the simulation where the exact pixel height is known. Another reason why the initial densities may differ is due to the polymer coating around the nanoparticles used for the experiments. This

creates a mix of materials which is not present in the simulations. The polymer coating affects the measurements of the density before sintering, as the coating decays before sintering starts. Within the uncertainty the initial densities can be said to agree and as such the simulations can be calibrated against the experiments

5.4 Calibration

The calibration of the simulation to experimental data was carried out, and the results are discussed below. The set of constants which gave the closest results found for calibrating the simulation to the experiments are listed in Table 5.2. The constants in this table correspond to the 'sintering parameters' explained in Chapter 2.

	Sintering Parameters								Calibration
Sno	conA	conB	conCgbe	congbenergy	consurfenergy	congbdif	consurfdif	convoldif	Temperature (˚C)
502	10.24	4.1	7	6	29.85	5.57	55.7	0.0557	450
505	8.15	4.1	7	6	24.74	6.65	66.5	0.0665	500
504	8.03	4.1	7	6	23.12	7.13	71.3	0.0713	550

Table 5.2: Sintering parameters corresponding to different temperatures

The results for the time calibration values are shown in Table 5.3 and Figure 5.26. The calibration factor was determined through the MATLAB minimization algorithm discussed in Chapter 4.

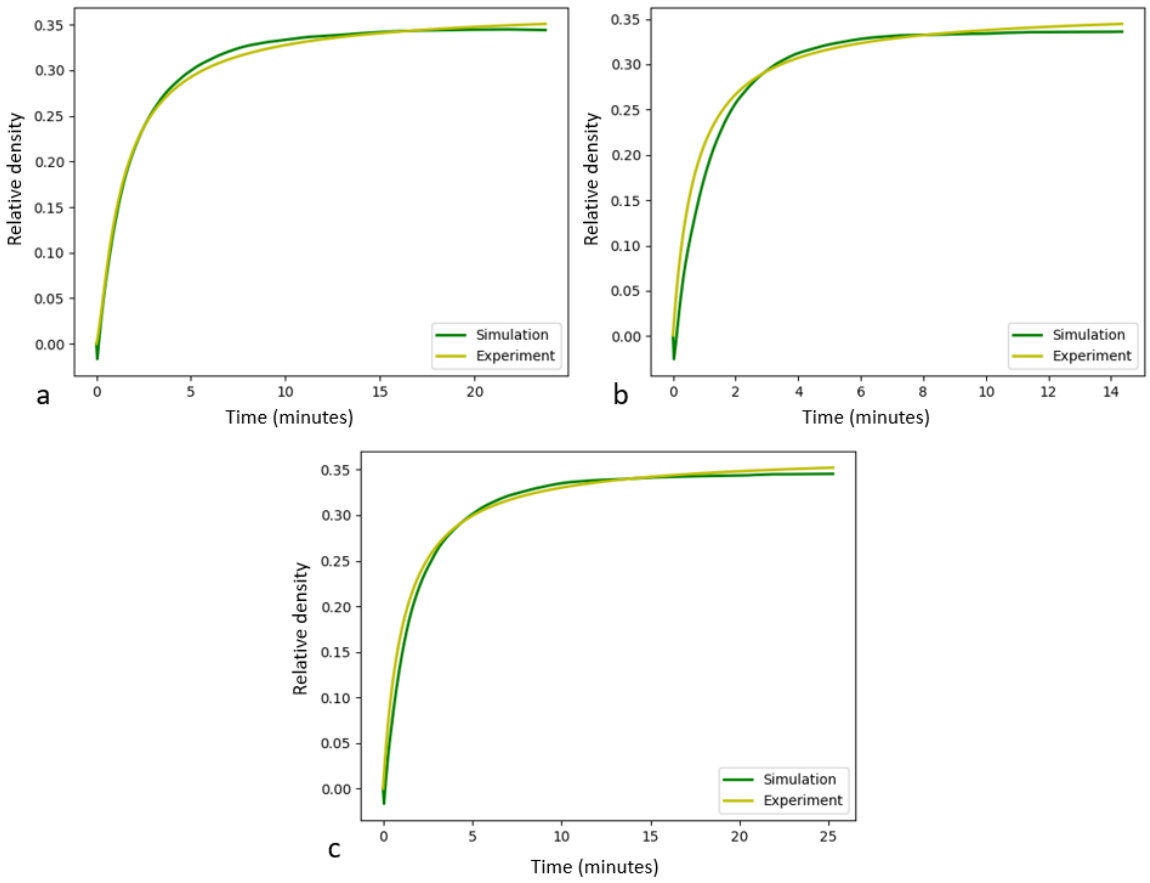


Figure 5.26: Comparing experimental fit to simulations for a) 450˚C. b) 500˚C. c) 550˚C

The results of fitting the simulation to experiments are shown in Figure 5.26. Table 5.3 shows the results for the range and averages of the error and time calibration values. The range included in these values, like the error bars in Figure 5.12, are a result of carrying out the calibration analysis on different boxes in the center of the simulation bed. The data in Table 5.3 shows an overall average error of about 9% between the simulation and experiments. The time calibration factor shown in the table has an average of 20% deviation between each temperature.

	Time Calibration, A (timesteps/minute)			Error (%)		
	Minimum	Maximum	Average	Minimum	Maximum	Average
450˚C	101070	388864	215319	3.22	30.14	13.54
500˚C	57644	277293	167598	3.57	10.91	7.01
550˚C	71110	383773	250102	3.99	9.79	6.76

Table 5.3: Calibration Results: Time and Error

Once calculated, the simulation diffusion constants were mapped using Eq. 32.

$$D_{sim}\left(\frac{pixels^2}{timesteps}\right) => \frac{D_{sim}\,A}{60S^2} => \; D_{actl}\left(\frac{cm^2}{s}\right) \tag{32}$$

Where D_{sim} represents the simulation diffusion coefficients and D_{actl} represents the diffusion coefficients when mapped to the corresponding units in a CGS system. A is the time calibration constant in Eq. 30 having units of (time steps)/minute and S is the size calibration constant. S here is set as 944822 pixels/cm. The values used for the diffusion constants in the simulation were mapped to physical units and are shown in Table 5.4. For this study surface, grain boundary and volume diffusion coefficients are assumed to follow the ratio 1000:100:1 [33,34], and surface energy and grain boundary energy approximately follows the ratio 2:1. These ratios were chosen based off literature and the analysis done with varying the properties in the 2D simulation. Future work still must be done to expand on the ratio of surface to grain boundary energy used and compare the results to the uncertainty of the simulation. As these constants are the final constants derived from the best fit between the experiments and the simulation, they represent sintering simulation constants for copper at the temperatures stated in the table.

	Surface Diffusion coefficients (cm^2/s)	
	Average	Uncertainty
450˚C	2E-07	1E-07
500˚C	2.2E-07	7.9E-08
550˚C	3E-07	1E-07

Table 5.4. Calibration Results: Diffusion constants

As of this point there has been no experimentally determined surface diffusion coefficients for copper nanoparticles to compare against the constants in Table 5.4. However, Bonzel and Gjostein [35] found that at 500˚C the surface diffusion coefficient of bulk copper is 1.91×10^{-7} cm^2/s. This value falls within the same order of magnitude as the value obtained from the simulations, which is further validation of the model.

CHAPTER 6: CONCLUSION AND FUTURE WORK

The conclusions of this thesis are presented and discussed in this chapter. After the conclusion, the future steps for the simulation are outlined and explained.

6.1 Conclusion

In this thesis a Phase Field Modelling (PFM) approach is used to simulate the sintering of particles in a one-by-one micrometer bed. The simulations presented here are unique in that they are applied to 3D bed scale simulations using clusters which can be extended to hundreds of particles. The process of preparing and running these simulations is described in detail. Common outputs from running these simulations are presented, and the outline from setting up the simulation to getting the analyzed results is presented. The validity of expanding this simulation is tested against a larger two-by-two micrometer bed containing 134 particles. Additionally, a data analysis package was created that measures the relative density change in each bed. Experiments were also performed with a TGA and a furnace. The relative advantages and accuracy between both of these methods were compared and presented and the furnace was chosen for the final sintering experiments. The experiments were performed over the range between 450 and 600˚C, heating up to 45 minutes. The experimental data shows that after 10 minutes of heating at 450˚C the density value starts to reach the maximum steady state value. As heating temperature increases, the amount of time the nanoparticles take to reach steady state decreases. The densification curves obtained from the experimental data are calibrated to match sintering simulation results and it is seen that the simulation trend is in good agreement with the experimental data. These comparisons between the simulation and experiments gave a calibration factor mapping the simulation time step to sintering time in minutes. The time calibration factor

derived was then used to map the diffusion constants from simulation units to physical units and the resulting surface diffusion constant showed good agreement with experimental data.

6.2 Future work

Some next steps for the simulation include uncertainty quantification, transitioning to gradient heating, including multiple layers, and creating a large bed model. The motivations for these improvements are discussed in this section.

6.2.1 Uncertainty Quantification

Given the simulation work and experiments carried out to calibrate the simulation, the next step is to quantify the error in the simulation. As shown in the uncertainty in changing the position of the analysis box, there is a significant amount of error that comes with the configuration of the particles in the bed. To completely quantify this uncertainty, different simulation beds have to be tested to determine the changes in calibration constants and rate of densification with changing the initial configuration of the bed.

6.2.2 Gradient heating

Next, given the map of simulation constants to physical temperature, the simulation would have to be transitioned to include transient heating more representative of a temperature gradient. This temperature gradient is important because in a typical laser sintering process, there is a temperature difference between the particles in the laser irradiated spot and the surrounding particles outside the laser heating zone. One way to model this gradient heating would be to combine a laser profile simulation with this. If a laser profile simulation is able to predict the temperature of particles in an irradiated laser

spot, as well as the rate of dissipation of that thermal energy (temperature in surrounding particles) then those temperatures can be mapped to the calibrated simulation constants. These constants can then be applied to the accurate particles in order to simulate gradient heating.

6.2.3 MULTIPLE LAYERS

Also, the simulation has to be transitioned to include multiple layers. This would enable the study of the effects that the interaction between layers has on the properties of the sintered bed. Multiple layers for the simulation could possibly be generated by combining the sintering simulation with the bed generation simulation. In this case, the top layer of the final sintered bed becomes the base in the bed generation simulation, and the additional layer is formed on top of this.

6.2.4 MILLIMETER BEDS CONTINUUM MODEL

This thesis has shown the ability to model hundreds of particles with the simulations introduced in this project. The information derived from these simulations gives a detailed representation of the microstructural evolution of micron sized beds undergoing sintering. However, expanding this simulation to a bed scale where the substrate size is in the millimeter range is impossible to do in any reasonable time. As such, a future step of the simulation is to extract information, such as the densification and shrinkage, from the detailed microstructural evolution in these simulations and use this information to inform a large bed continuum model. This will allow for simulations of the sintering of millimeter sized particles with a level of detail which is not possible using only continuum models.

REFERENCES

[1] Govett, T., Kim, K., Lundin, M. and Pinero, D., 2012, "Design Rules for Selective Laser Sintering," Mechanical Engineering Design Projects Program, The University of Texas at Austin, Austin, TX. https://www.me.utexas.edu/~ppmdlab/files/designers.guide.sls.pdf

[2] Sager B. and Rosen D., 2002, "Stereolithography Process Resolution," Annual International Solid Freeform Fabrication Symposium, pp. 575-584.

[3] Nelson, C., McAlea, K., and Gray, D., 1995, "Improvements in SLS Part Accuracy," Annual International Solid Freeform Fabrication Symposium, pp. 159-169.

[4] Roy, N. K., Foong, C. S., and Cullinan, M. A., 2016, "Design of a Micro-Scale Selective Laser Sintering System," 2016 Annual International Solid Freeform Fabrication Symposium, pp. 1495-1508.

[5] Roy, N., Yuksel, A., and Cullinan, M., 2016, "Design and Modeling of a Microscale Selective Laser Sintering System," ASME 2016 11th International Manufacturing Science and Engineering Conference.

[6] Roy, N., Dibua, O., Foong, C. S., and Cullinan, M., 2017, "Preliminary Results on the Fabrication of Interconnect Structures Using Microscale Selective Laser Sintering," Proceedings of the ASME 2017 International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems.

[7] Dong, L., Makradi, A., Ahzi, S., and Remond, Y., 2009, "Three-dimensional transient finite element analysis of the selective laser sintering process," Journal of Materials Processing Technology, 209, pp. 700-706.

[8] Moser, D., Cullinan, M., and Murthy, J., 2016, “Particle-Scale Melt Modeling of the Selective Laser Melting Process,” 2016 Annual International Solid Freeform Fabrication Symposium, pp. 247-256.

[9] Moser, D., Fish, S., Beaman, J., and Murthy, J., 2014, “Multi-Layer Computational Modeling of Selective Laser Sintering Processes,” ASME 2014 International Mechanical Engineering Congress and Exposition, 2A, pp. V02AT02A008-V02AT02A018.

[10] Ko, S. H., and Grigoropoulos, C. P., 2008, “The Solid-State Neck Growth Mechanisms in Low Energy Laser Sintering of Gold Nanoparticles: A Molecular Dynamics Simulation Study,” Journal of Heat Transfer, 130, pp. 092404-1-092404-7.

[11] Rojek, J., Nosewicz, S., Pietrzak, K., and Chmielewski, M., 2013, “Simulation of Powder Sintering Using a Discrete Element Model,” Acta Mechanica et Automatica, 7(3), pp. 175-179.

[12] Liu, X., Schnell, K. S., Simon, J., Kruger, P., Bedeaux, D., Kjelstrup, S., Bardow, A., and Vlugt, J. H. T., 2013, “Diffusion Coefficients from Molecular Dynamics Simulations in Binary and Ternary Mixtrures,” Int J Thermophys, 34, pp. 1169-1196.

[13] Ding, L., Davidchack, L. R., and Pan, J., 2009, “A molecular dynamics study of sintering between nanoparticles,” Computational Materials Science, 45, pp. 247-256.

[14] Li, F., and Pan, J., “Modelling “Nano-Effects” in Sintering”. Full text: Sintering: Mechanisms of Convention Nanodensification and Field Assisted Processes (Springer: 2013) pp. 17-34.

[15] Cheng, B., and Ngan, H. W. A., 2013, “The Sintering and Densification Behavior of Many Copper Nanoparticles: A molecular Dynamics Study,” Computational Materials Science, 74, pp. 1-11.

[16] Braginsky, M., Tikare, V., and Olevsky, E., 2005, “Numerical simulation of solid state sintering”, International Journal of Solids and Structures, 42 (2), pp. 621-636.

[17] Chen, S., Xu, Y., and Jiao, Y., 2016, “Modelling morphology evolution and densification during solid-state sintering via kinetic Monte Carlo simulation”, Modelling Simul. Mater. Sci. Eng., 24(8) pp. 085003-0850017.

[18] Bjork, R., Tikare, V., Frandsen, H. L., and Pryds, N., “The effect of particle size distributions on the microstructural evolution during sintering”, J. Am. Ceram. Soc., 96(1) pp. 103-110.

[19] Wang, U. Y., 2006, “Computer Modeling and Simulation of Solid-State Sintering: A Phase Field Approach,” Acta Materialia, 54, pp. 953-961.

[20] Shinagawa, K., 2014, “Simulation of Grain Growth and Sintering Process by Combined Phase-Field/Discrete Element Method,” Acta Materialia, 66, pp. 360-369.

[21] Kumar, V., 2011, “Simulations and Modeling of Unequal Sized Particles Sintering,” Doctor of Philosophy Dissertation, The University of Utah, Salt Lake City, UT.

[22] Chockalingam, K., Kouznetsova, V., van der Sluis, O., and Geers, M. G. D., 2016, “2D phase field modeling of sintering of silver nanoparticles”, Computer Methods in Applied Mechanics and Engineering, 312, pp. 492-508.

[23] Tikare, V., Holm, E. A., Fan, D., and Chen, L. Q., 1999, “Comparison of Phase-Field and Potts Models for Coarsening Processes”, Acta mater., 47(1) pp. 363-371.

[24] Moelans, N., Blanpain, B., and Wollants, P., 2008, “An introduction to phase-field modeling of microstructure evolution,” Computer Coupling of Phase Diagrams and Thermochemistry, 32, pp. 268-294.

[25] Cahn, J. W., and Hilliard, J. E., 1958, “Free Energy of a Nonuniform System. I. Interfacial Free Energy,” The Journal of Chemical Physics, 28, pp. 258-267.

[26] Cahn, J. W., 1961, “On Spinodal Decomposition,” Acta Metall., 9(9), pp. 81–87. https://doi.org/10.1016/0001-6160(61)90182-1

[27] Ginzburgh, V. L., and Landau, L. D., 1950, “On the Theory of Superconductivity,” Zh. Eksperim. I Teor. Fiz. 20, pp. 1064–1082. English translation in: L. D. Landau, Collected papers (Oxford: Pergamon Press, 1965) pp. 546-568. https://doi.org/10.1016/B978-0-08-010586-4.50078-X

[28] “Slurm Configuration File”, SchedMD. https://slurm.schedmd.com/slurm.conf.html

[29] n.d., 2018, “Stampede2 User Guide” TACC. https://portal.tacc.utexas.edu/user-guides/stampede2

[30] Yuksel, A., and Cullinan, M., 2016, “Modeling of nanoparticle agglomeration and powder bed formation in microscale selective laser sintering systems,” Additive Manufacturing, 12, pp. 204-215.

[31] Clark, I., 2017, “CI-005_Data_Sheet_2017,” Intrinsiq Materials, Hampshire, United Kingdom. http://intrinsiqmaterials.com/wp-content/uploads/2018/04/CI-005_Data_Sheet_2017_v1.3.pdf

[32] Giunta, A. A., and Watson T. L., “A comparison of approximation modeling techniques: polynomial versus interpolating models”, American Institute of Aeronautics and Astronautics, paper 98-4758.

[33] Deng, J., 2012, “A Phase Field Model of Sintering with Direction-Dependent Diffusion,” Materials transactions, 53, pp. 385-389.

[34] Zhang, D., Weng, G., Gong, S., and Zhou, D., 2003, “The Kinetics of Initial Stage in Sintering Process of BaTiO3-based PTCR ceramics and its computer simulation,” Materials Science & Engineering B, 99, pp. 88-92.

[35] Hoehne, K., and Sizmann, R., 1971, “Volume and Surface Self-Diffusion Measurements on Copper by Thermal Surface Smoothing,” Physica Status Solidi, 5, pp. 577-589.